B. Tech. in Computer Science and Engineering with Specialization in User Interface and User Experience (UI/UX) Design (2024 Admitted Batch)

FIRST SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	HSMC		English	2	0	0	2	2
2	HSMC		English Lab	0	0	2	2	1
3	BSC		Engineering Mathematics -1	3	1	0	4	4
4	BSC		Engineering Physics-I	3	1	0	4	4
5	BSC		Engineering Physics Lab-I	0	0	2	2	1
6	ESC		Problem Solving and Programming	3	0	0	3	3
7	ESC		Problem Solving and Programming Lab	0	0	2	2	1
8	ESC		Workshop Practices OR	0	0	3	3	1.5
9	ESC		Engineering Graphics	0	0	3		
10	PR		Project-I	0	0	2	2	1
11	MNC		UHV-I Mandatory Induction Program	2 Weeks				0
			Total				**24**	**18.5**

SECOND SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	HSMC		UHV II: Understanding Harmony	2	1	0	3	3
2	BSC		Engineering Mathematics -II	3	1	0	4	4
3	BSC		Engineering Physics-II	3	0	0	3	3
4	BSC		Engineering Physics Lab - II	0	0	2	2	1
5	ESC		Basic Electrical Engineering	3	1	0	4	4
6	ESC		Basic Electrical Engineering Lab	0	0	2	2	1
7	ESC		Data Structures and Algorithms	3	0	0	3	3
8	ESC		Data Structures and Algorithms Lab	0	0	4	4	2
9	ESC		Workshop Practices OR	0	0	3	3	1.5
10	ESC		Engineering Graphics	0	0	3		
11	PR		Project-II	0	0	2	2	1
12	MNC		Professional Communication Practice (AUDIT)	0	1	0	1	0
			Total				**31**	**23.5**

THIRD SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	BSC	24XXXXXX	Mathematical Foundations for Artificial Intelligence and Data Science	3	1	0	4	4
2	PCC	XXXXXXX	Theory of Computation	3	0	0	3	3
3	PCC	XXXXXXX	Object Oriented Systems and Programming	3	0	0	3	3
4	PCC	XXXXXXX	Object Oriented Systems and Programming Lab	0	0	4	4	2
5	PCC	XXXXXXX	Database Management Systems	3	0	0	3	3
6	PCC	XXXXXXX	Database Management Systems Lab	0	0	2	2	1
7	PCC	XXXXXXX	Unix Programming Lab	1	0	2	3	2
8	PCC	XXXXXXX	Object Oriented Programming using Java	0	0	2	2	1
9	HSC	XXXXXXX	Economics	2	1	0	3	3
10	PRC	XXXXXXX	Summer Training-I (4 weeks)*	0	0	0	0	2
11	PCC	XXXXXXX	Competitive Programming-I	0	0	2	2	1
			Total				**29**	**25**
Specialization in User Interface and User Experience (UI/UX) Design								
12	PCC	XXXXXXX	Introduction to UI/UX	3	0	0	3	3
13	PCC	XXXXXXX	Introduction to UI/UX Lab	0	0	2	2	1
			Total				**34**	**29**

FOURTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	HSC	XXXXXXX	HSS Elective	2	1	0	3	3
2	PCC	XXXXXXX	Digital Systems and Computer Organisation	3	0	0	3	3
	PCC	XXXXXXX	Digital Systems and Computer Organisation Lab	0	0	2	2	1
3	PCC	XXXXXXX	Design and Analysis of Algorithms	3	1	0	4	4
4	PCC	XXXXXXX	Design and Analysis of Algorithms Lab	0	0	2	2	1
5	PCC	XXXXXXX	Artificial Intelligence and Machine Learning	3	0	0	3	3
6	PCC	XXXXXXX	Artificial Intelligence and Machine Learning Lab	0	0	2	2	1
7	PCC	XXXXXXX	Software Engineering	3	0	0	3	3
8	PCC	XXXXXXX	Competitive Programming-II	0	0	2	2	1
9	PEC	XXXXXXX	Discipline Elective – 1	2	0	0	2	2
10	PEC	XXXXXXX	Discipline Elective – 1 Lab	0	0	2	2	1
11	OMC	XXXXXXX	Environmental Studies	3	0	0	3	Qualifying
			Total				**31**	**23**
Specialization in User Interface and User Experience (UI/UX) Design								
12	PCC	XXXXXXX	Visual and Interaction Design	3	0	0	3	3
13	PCC	XXXXXXX	Visual and Interaction Design Lab	0	0	2	2	1
			Total				**36**	**27**

FIFTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PCC	XXXXXXX	Operating Systems	3	0	0	3	3
2	PCC	XXXXXXX	Operating Systems Lab	0	0	2	2	1
3	PCC	XXXXXXX	Computer Networks	3	1	0	4	4
4	PCC	XXXXXXX	Computer Networks Lab	0	0	2	2	1
5	PCC	XXXXXXX	Full Stack Development Lab	0	0	2	2	1
6	PEC	XXXXXXX	Discipline Elective – 2	2	0	0	2	2
7	PEC	XXXXXXX	Discipline Elective – 2 Lab	0	0	2	2	1
8	PEC	XXXXXXX	Discipline Elective – 3	2	0	0	2	2
9	PEC	XXXXXXX	Discipline Elective – 3 Lab	0	0	2	2	1
10	OMC	XXXXXXX	Indian Constitution & Traditional Knowledge	3	0	0	3	Qualifying
11	PRC	XXXXXXX	Summer Training-II (6 weeks) *	0	0	0	0	2
12	PCC	XXXXXXX	Competitive Programming-III	0	0	2	2	1
13	HSC	XXXXXXX	Logical and Quantitative Techniques-I	2	0	0	2	2
			Total				**28**	**21**
Specialization in User Interface and User Experience (UI/UX) Design								
14	PCC	XXXXXXX	Advanced Wireframing, Prototyping and Responsive Design	3	0	0	3	3
15	PCC	XXXXXXX	Advanced Wireframing, Prototyping and Responsive Design Lab	0	0	2	2	1
			Total				**33**	**25**

SIXTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PCC	XXXXXXX	Web Technology	3	0	0	3	3
2	PCC	XXXXXXX	Web Technology Lab	0	0	2	2	1
3	PCC	XXXXXXX	Advanced Data Structures and Algorithms	3	0	0	3	3
4	PCC	XXXXXXX	Advanced Data Structures and Algorithms Lab	0	0	2	2	1
5	PCC	XXXX FLEXI CORE	Distributed and Cloud Computing **OR** Information Security and Cryptography	3	0	0	3	3
6	PEC	XXXXXXX	Discipline Elective – 4	3	0	0	3	3
7	PEC	XXXXXXX	Discipline Elective – 5	3	0	0	3	3
8	OEC	XXXXXXX	Open Elective – 1	3	0	0	3	3
9	Value Added	XXXXXXX	Selected Value-Added Course	2	0	0	2	Audit
10	HSC	XXXXXXX	Soft Skills for Employability	0	0	2	2	1
11	PRC	XXXXXXX	Minor Project	0	0	4	4	2
12	HSC	XXXXXX	Logical and Quantitative Techniques-II	2	0	0	2	2
			Total				**32**	**25**
Specialization in User Interface and User Experience (UI/UX) Design								
13	PCC	XXXXXXX	UI/UX Tools and Technologies	3	0	0	3	3
14	PCC	XXXXXXX	Portfolio Building and Freelancing	3	0	0	3	3
			Total				**38**	**31**

SEVENTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PEC	XXXXXXX	Discipline Elective – 6	3	0	0	3	3
2	OEC	XXXXXXX	Open Elective – 2	3	0	0	3	3
3	PRC	XXXXXXX	Major Project Part – 1	0	0	0	8	4
4	PRC	XXXXXXX	Summer Training - III (6 weeks) *	0	0	0	0	4
			Total				**14**	**14**

EIGHTH SEMESTER

S. No.	Course			Contact Hours				Credits
	Category	Course Code	Course Title	L	T	P	Total	
1	PEC	XXXXXXX	Discipline Elective –7	3	0	0	3	3
2	OEC	XXXXXXX	Open Elective –3	3	0	0	3	3
3	PRC	XXXXXXX	Major Project Part –2	0	0	0	16	8
			Total				**22**	**14**

***To be completed during the summer vacation.**

Total Program Credits: 18.5 + 23.5 + 29 + 27 + 25 + 31 + 14 + 14 = 182

Discipline Elective – 1 and Discipline Elective – 1 Lab (offered in 4^{th} Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Data Analytics using R and Python	2	0	0	2	2
2	XXXXXXX	Data Analytics using R and Python Lab	0	0	2	2	1
3	XXXXXXX	Mobile Application Development	2	0	0	2	2
4	XXXXXXX	Mobile Application Development Lab	0	0	2	2	1
5	XXXXXXX	Fundamentals of Smart Systems and IoT	2	0	0	2	2
6	XXXXXXX	Fundamentals of Smart Systems and IoT Lab	0	0	2	2	1
7	XXXXXXX	Introduction to Compiler Design	2	0	0	2	2
8	XXXXXXX	Introduction to Compiler Design Lab	0	0	2	2	1
9	XXXXXXX	Data Engineering	2	0	0	2	2
10	XXXXXXX	Data Engineering Lab	0	0	2	2	1
11	XXXXXXX	Foundation of Data Science and Visualization	2	0	0	2	2
12	XXXXXXX	Foundation of Data Science and Visualization Lab	0	0	2	2	1
13	XXXXXXX	Artificial and Computational Intelligence	2	0	0	2	2
14	XXXXXXX	Artificial and Computational Intelligence Lab	0	0	2	2	1

Discipline Elective – 2 and Discipline Elective – 2 Lab (offered in 5th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Fundamentals of Soft Computing	2	0	0	2	2
2	XXXXXXX	Fundamentals of Soft Computing Lab	0	0	2	2	1
3	XXXXXXX	Fundamentals of Computer and Cyber Security	2	0	0	2	2
4	XXXXXXX	Fundamentals of Computer and Cyber Security Lab	0	0	2	2	1
5	XXXXXXX	Data Mining and Data Warehousing	2	0	0	2	2
6	XXXXXXX	Data Mining and Data Warehousing Lab	0	0	2	2	1
7	XXXXXXX	Agile Software Development Process	2	0	0	2	2
8	XXXXXXX	Agile Software Development Process Lab	0	0	2	2	1
9	XXXXXXX	IoT Analytics	2	0	0	2	2
10	XXXXXXX	IoT Analytics Lab	0	0	2	2	1
11	XXXXXXX	Big Data Analytics	2	0	0	2	2
12	XXXXXXX	Big Data Analytics Lab	0	0	2	2	1
13	XXXXXXX	Search in AI	2	0	0	2	2
14	XXXXXXX	Search in AI Lab	0	0	2	2	1
15	XXXXXXX	Intelligent Robotics and Autonomous Systems	2	0	0	2	2
16	XXXXXXX	Intelligent Robotics and Autonomous Systems Lab	0	0	2	2	1

Discipline Elective – 3 and Discipline Elective – 3 Lab (offered in 5th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Image Processing and Computer Vision	2	0	0	2	2
2	XXXXXXX	Image Processing and Computer Vision Lab	0	0	2	2	1
3	XXXXXXX	Introduction to Blockchain Technology	2	0	0	2	2
4	XXXXXXX	Introduction to Blockchain Technology Lab	0	0	2	2	1
5	XXXXXXX	Computing for Data Science	2	0	0	2	2
6	XXXXXXX	Computing for Data Science Lab	0	0	2	2	1
7	XXXXXXX	Sensor Technology & Android Programming	2	0	0	2	2
8	XXXXXXX	Sensor Technology & Android Programming Lab	0	0	2	2	1
9	XXXXXXX	Concept of Graph Theory	2	0	0	2	2
10	XXXXXXX	Concept of Graph Theory Lab	0	0	2	2	1
11	XXXXXXX	Computational Data Analysis	2	0	0	2	2
12	XXXXXXX	Computational Data Analysis Lab	0	0	2	2	1
13	XXXXXXX	Embedded Software Development	2	0	0	2	2
14	XXXXXXX	Embedded Software Development Lab	0	0	2	2	1

Discipline Elective – 4 (offered in 6th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Introduction to Deep Learning	3	0	0	3	3

2	XXXXXXX	Cryptocurrency Technologies	3	0	0	3	3
3	XXXXXXX	Information Retrieval and Semantic Web	3	0	0	3	3
4	XXXXXXX	Cloud Computing Essentials: Azure and AWS	3	0	0	3	3
5	XXXXXXX	Reinforcement Learning	3	0	0	3	3
6	XXXXXXX	Responsible AI and AI Ethics	3	0	0	3	3

Discipline Elective – 5 (offered in 6th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Machine Learning and Big Data	3	0	0	3	3
2	XXXXXXX	Secure Design of Software Systems	3	0	0	3	3
3	XXXXXXX	Fog and Edge Computing	3	0	0	3	3
4	XXXXXXX	Statistical Analysis and Computing	3	0	0	3	3
5	XXXXXXX	Time Series Analysis and Forecasting	3	0	0	3	3
6	XXXXXXX	Edge and Federated Learning	3	0	0	3	3
7	XXXXXXX	Retrieval Augmented Generation	3	0	0	3	3

Discipline Elective – 6 (offered in 7th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Machine Learning and Natural Language Processing	3	0	0	3	3
2	XXXXXXX	Ethical Hacking & Prevention	3	0	0	3	3
3	XXXXXXX	Introduction to Large Scale Database Systems	3	0	0	3	3
4	XXXXXXX	Introduction to DevOps	3	0	0	3	3
5	XXXXXXX	Industrial Automation and IOT	3	0	0	3	3
6	XXXXXXX	Explainable AI	3	0	0	3	3
7	XXXXXXX	Probabilistic Graphical Models	3	0	0	3	3
8	XXXXXXX	AI for IoT	3	0	0	3	3
9	XXXXXXX	Sentiment Analysis and Opinion Mining	3	0	0	3	3

Discipline Elective – 7 (offered in 8th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	AI for Healthcare & Smart Systems	3	0	0	2	3
2	XXXXXXX	Digital Forensics and Cyber Laws	3	0	0	2	3
3	XXXXXXX	Social Network Analysis	3	0	0	2	3
4	XXXXXXX	Software Construction using Kubernetics and Micro-services	3	0	0	3	3
5	XXXXXXX	AI for Finance	3	0	0	3	3
6	XXXXXXX	Agentic AI	3	0	0	3	3

Open Elective – I (offered in 6th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Cyber Security and Digital Safety	3	0	0	3	3
2	XXXXXXX	Business Intelligence and Analytics	3	0	0	3	3

Open Elective – 2 (offered in 7th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	User Interface and User Experience Design	3	0	0	3	3
2	XXXXXXX	Software Testing Methodologies	3	0	0	3	3

Open Elective – 3 (offered in 8th Semester)

S. No.	Course Code	Course Title	L	T	P	Total	Credits
1	XXXXXXX	Social Media Analytics	3	0	0	3	3
2	XXXXXXX	Digital Twin - Fundamental Concepts to Applications in Advanced Manufacturing	3	0	0	3	3

Problem Solving and Programming

COURSE CODE: 24B11CI111

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: NONE

Course Objectives:

1. To formulate simple algorithms for arithmetic and logical problems.
2. To translate the algorithms to programs (in C language).
3. To test and execute the programs and correct syntax and logical errors.
4. To implement conditional branching, iteration and recursion.
5. To understand various number systems associated with C.
6. To decompose a problem into functions and synthesize a complete program using divide and conquer approach.
7. To use arrays, pointers and structures to formulate algorithms and programs.
8. To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
9. To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To formulate simple algorithms for arithmetic and logical problems.	Familiarity
CO-2	To translate the algorithms to programs (in C language).	Assessment
CO-3	To understand various number systems associated with C	Usage
CO-4	To test and execute the programs and correct syntax and logical errors.	Usage
CO-5	To implement conditional branching, iteration and recursion.	Usage
CO-6	To use arrays, pointers and structures to formulate algorithms and programs.	Usage
CO-7	To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.	Usage

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Programming, Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.) Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples. (1	**4**

	lecture) From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable	
2	Number System: Decimal, Binary, Octal, Hexadecimal and interconversion.	**3**
3	Arithmetic expressions and precedence	**3**
4	Conditional Branching and Loops, Writing and evaluation of conditionals and consequent branching	**6**
5	Arrays (1-D, 2-D), Character arrays and Strings	**4**
6	Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs	**4**
7	Function: Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference. Recursion: Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort and Merge sort. Tracking Recursion	**8**
8	Idea of pointers, defining pointers, Use of Pointers in self-referential structures.	**4**
9	Structures, Defining structures and Array of Structures, notion of linked list (Basic Idea of Implementation).	**2**
10	File Handling	**4**
Total lectures		**42**

Suggested Text Book(s):

1. Mittal, Ajay. Programming in C: A Practical Approach. Pearson Education India, 2010.
2. Kernighan, Brian W., and Dennis M. Ritchie. "The C programming language." (2002).

Suggested Reference Book(s):

1. Deitel, H., and PJ C+ Deitel. How to program. Pearson, 2021.

Other useful resource(s):

1. Link to NPTEL course contents: https://www.youtube.com/watch?v=t9WKOcRB63Q&list=PLJ5C_6qdAvBFzL9su5J-FX8x80BMhkPy1

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-6	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-7	2	3	3	3	3	1	1	1	2	2	1	**2**
Average	**2**	**2.57**	**2.57**	**2.57**	**2.57**	**1**	**1**	**1**	**2**	**2.14**	**1.43**	

Problem Solving and Programming Lab

COURSE CODE: 24B17CI171

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Fundamentals of C Programming and Algorithms

Course Objectives:

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understanding C program development, Compilation, debunking, running, etc.	Familiarity
CO-2	Introduction to C Programming using Control Statements and Repetition Statement	Usage
CO-3	Apply and practice logical formulations to solve some simple problems leading to specific problems.	Assessment
CO-4	Design effectively the required programming components that efficiently solve computing problems in the real world.	Usage

List of Experiments

S. No	Description	Hours
1	Getting acquainted with the C program Structure and basic Functions, Getting acquainted with the various data types and arithmetic operators in C	**2**
2	Write a program to obtain the reversed number and to determine whether the original and reversed numbers are equal or not.	**2**
3	Write a program to check whether a triangle is valid or not, when the three angles of triangle are entered through the keyboard. A triangle is valid if the sum of all three angles is equal to 180 degrees.	**2**
4	Check a given I/P is character, number or special symbol.	**2**
5	WAP to check if a given number is Armstrong or not. Calculate factorial of a number, Given number is prime or not.	**2**
6	a) Write a program to add the first seven terms of the following series using any loop: 1/1! + 2/2! + 3/3! +..... b) Develop start patterns using loops	**4**
7	Swap two numbers via various methods.	**2**
8	Functions and Recursion Applications: Factorial, sum, etc.	**2**

9	Programs on Arrays: searching and sorting	**3**
10	Dynamic Memory Allocation and Structures	**2**
11	File Handling	**2**
Total Lab hours		**25**

Suggested/Resources:

1. Deitel and Deitel, C How to Program, 7° Edition, 2013.
2. Venugopal Prasad, Mastering C, Tata McGraw Hill.
3. Complete Reference with C, Tata McGraw Hill.
4. Drmey, How to solve it by Computer, PHI.
5. Kerninghan and Ritchie, The C Programming Language.
6. http://www.acm.uiuc.edu/webmonkeys/book/c_guide/
7. http://msdn.microsoft.com/en-us/library/25db87se.aspx

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evalution	20
2	End Sem. Evalution	20
3.	Attendance	15
4.	Lab Assessment	45
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	Average
CO-1	3	3	3	3	2	2	1	1	1	1	1	1.91
CO-2	3	3	3	3	3	1	1	1	1	1	1	1.91
CO-3	3	3	2	3	2	3	2	1	1	1	2	2.09
CO-4	3	3	3	2	3	2	1	1	1	1	1	1.91
Average	3	3	2.75	2.75	2.5	2	1.25	1	1	1	1.25	

Data Structure and Algorithms

COURSE CODE: 24B11CI211

COURSE CREDIT: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisites: C/C++

Course Objectives:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures
5. Introduce students to data abstraction and fundamental data structures.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To gain knowledge on the notions of data structure, Abstract Data Type.	Familiarity
CO-2	For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.	Assessment
CO-3	For a given Search problem (Linear Search and Binary Search) student will able to implement it.	Assessment
CO-4	For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.	Assessment
CO-5	Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.	Assessment
CO-6	Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Searching: Linear Search and Binary Search Techniques and their complexity analysis.	**7**
2	**Stacks:** ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis.	**5**
3	**Queues:** ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.	**5**

4	**Linked Lists:** Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.	**8**
5	**Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.	**6**
6	**Sorting and Hashing:** Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.	**6**
7	**Graph:** Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.	**5**
Total lectures		**42**

Suggested Text Book(s):

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press

Suggested Reference Book(s):

1. "Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. "How to Solve it by Computer", 2nd Impression by R. G. Dromey, Pearson Education.
3. "Data structures and Algorithms Made Easy" 5th edition by Narasimha Karumanchi, Career monk publications
4. "Data Structureand Algorithms in C" 2ndedition by Mark Allen Weiss (2002), Pearson Education

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106102064/
2. Link to topics related to course:
   a. https://onlinecourses.nptel.ac.in/noc18_cs25/preview
   b. https://nptel.ac.in/courses/106103069/
   c. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course Outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	**2.36**
CO-2	3	3	3	2	3	2	3	2	2	3	1	**2.45**
CO-3	3	3	3	2	2	3	1	2	3	3	1	**2.36**
CO-4	3	3	3	2	3	3	2	2	3	3	1	**2.55**
CO-5	3	3	3	2	3	3	2	2	3	3	1	**2.55**
CO-6	3	3	3	2	3	3	2	2	2	3	1	**2.45**
Average	**3**	**3**	**3**	**2**	**2.67**	**2.83**	**2**	**2**	**2.5**	**3**	**1**	

Data Structure and Algorithms Lab

COURSE CODE: 18B17CI271

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

L-T-P: 0-0-4

Pre-requisites: C programming

Course Objectives:

1. Develop problem solving ability using Programming
2. Develop ability to design and analyze algorithms
3. Introduce students to data abstraction and fundamental data structures
4. Develop ability to design and evaluate Abstract Data Types and data structures
5. Apply data structure concepts to various examples and real life applications

Course outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To gain knowledge on the notions of data structure, Abstract Data Type	Familiarity
CO-2	To have hands on skills to evaluate different kinds of linked lists and their applications in day-to-day problem solving.	Usage
CO-3	To have hands on skills to evaluate different kinds stacks and their applications and implementations in day-to-day problem solving	Assessment
CO-4	To have hands on skills to evaluate different kinds queues and their applications and implementations in simulations.	Assessment
CO-5	To acquire knowledge of various sorting algorithms	Usage
CO-6	To learn Searching: Balanced tree, red-black tree, lower bounds for searching	Usage
CO-7	To learn to code for operations on Tree or BST (Creation; Traversing like pre-order, post-order and in-order; Searching element; finding height etc.)	Usage
CO-8	Introduction to Heaps	Usage
CO-9	To learn to code for operations on Graphs (Creation; entering info, printing output and deleting; traversal of BFS and DFS algorithm)	Assessment

List of Experiments:

S. No	Description	Hours
1	**Getting acquainted with** a) Arrays and Strings, Structures, b) Recursion, Pointers c) Dynamic memory allocation	 2 4 4
2	**Operations on: (Creation, insertion, deletion, sorting, traversing, reversing etc)**	

	a) Linear Linked List, b) Doubly and c) Circular Linked List	4 4 2
3	**Operations on Stacks:** a) Creation; pushing; popping; b) testing underflow, overflow; c) prefix and postfix	 4 2 2
4	**Operations on Queues:** a) Creation; b) enqueue; dequeue; c) testing underflow, overflow	 4 2 2
5	**Operations on Tree or BST:** Creation; a) Traversing like preorder, post-order and in-order; b) Searching element; finding height etc.	 4 2
6	**Implementation of sorting algorithms 1:** Insertion Sort and Selection Sort Algorithm with arrays using dynamic memory allocation.	2
7	**Implementation of sorting algorithms 2:** Bubble Sort and Merge Sort Algorithm with arrays using dynamic memory allocation.	2
8	**Implementation of sorting algorithms 3:** Implementation of Radix Sort and Quick Sort Algorithm with arrays using dynamic memory allocation.	2
9	**Operation on Heaps:** a) Heaps, b) Heap Sort	 2 2
10	**Implementation of Searching algorithms:** Linear Search Algorithm and Binary Search Algorithm using dynamic memory allocation.	2
11	**Operations on Graphs :** (Creation; entering info; printing Output and deleting; traversal of BFS and DFS algorithm etc.)	2
	Total Lab hours	**56**

Minor Project(s) – (*Only for 2 credit lab)*

- Design GUI based program to solve any binary equation.
- Design GUI based program to find the roots of quadratic equation.
- Design a program that picks the characters at equal interval from the given text/paragraph and generate a new paragraph in which each set of word can't have more than 4 characters. Last word of the paragraph can have <=4 characters.
- Program to input following data into disk file. Code, name, department and salary of employee in a firm. After creating file read the file and find following-
  Methodology
  algorithms
  Code execution
  Future scope
  Count number of employees as per department
  Search record of employee
  Display record of employee
  Display list of employee in alphabetical order as per department
  Read record from file

Suggested Books/Resources:

1. Langsam, Augestein, Tenenbaum : Data Structures using C and C++, 2nd Edn, 2000, Horowitz and Sahani : Fundamental of Data Structures in C, 2^{nd} Edn, 2008
2. Weiss : Data Structures and Algorithm Analysis in C/C++, 3rd Edn, 2006
3. Sahani : Data Structures, Algorithms and applications in C++, 1997.
4. Corman et al : Introduction to Algorithms, 3rd Edn., 2009
5. http://www.nptel.iitm.ac.in/video.php?subjectId=106102064, last accessed Mar 13, 2014.
6. http://www.cs.auckland.ac.nz/~jmor159/PLDS210/ds_ToC.html, last accessed Mar 13, 2014.
7. http://courses.cs.vt.edu/csonline/DataStructures/Lessons/index.html, last accessed Mar 13, 2014.
8. Link to topics related to course:
   a. http://cse.iitkgp.ac.in/~pallab/pds16/pds16.htm
   b. https://onlinecourses.nptel.ac.in/programming101/preview

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	2	2	3	3	3	2	3	2	**2.55**
CO-2	3	3	3	2	3	3	3	3	2	3	2	**2.73**
CO-3	3	3	3	2	2	3	3	3	3	3	2	**2.73**
CO-4	3	3	3	3	3	3	3	2	2	3	3	**2.82**
CO-5	3	3	3	2	2	2	3	3	3	3	2	**2.64**
CO-6	3	3	3	3	3	3	3	2	2	3	3	**2.82**
CO-7	3	3	3	3	3	3	2	2	3	3	3	**2.82**
CO-8	3	3	3	2	3	3	3	3	3	3	2	**2.82**
CO-9	3	3	2	3	3	3	3	3	3	2	3	**2.82**
Average	**3**	**2.89**	**2.89**	**2.44**	**2.67**	**2.89**	**2.89**	**2.67**	**2.56**	**2.89**	**2.44**	

Theory of Computation

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Basics of probability and statistics

Course Objectives:

1. To introduce the students to the mathematical foundations of computation including automata theory
2. To introduce the students about the theory of formal languages and grammars
3. To introduce the students about the theory of Automata which includes Finite Automata, Push down Automata and Turing Machines
4. To introduce the students about the notions of algorithm, decidability
5. To introduce the students about the Complexity and computability

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Broaden knowledge of the fundamental mathematical and computational principles that are the foundation of computer science	Understand
CO-2	Understand the concept of Deterministic Finite Automata and Non-Deterministic Finite Automata, minimize the states, usage Moore and Mealy Machine	Evaluate
CO-3	Understand how to use the context free grammars and how to derive parse trees and solve ambiguity problems; Understand Normal forms for Context Free Grammar's Chomsky and Greibach Normal Forms, Push Down Automaton algorithm	Evaluate
CO-4	Understand the basic concepts of Turing Machine, configuration of Turing Machine, computing, multiple tapes, two-way infinite tape concepts, concept of non-deterministic Turing machines	Apply
CO-5	Understand the computational power of languages, numerical functions applied to Turing machines, numerical functions applied to Turing machines, concept of halting problem. Grammars, properties of recursive languages, concept of polynomial decidable	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	General introduction, Complexity theory, Computability theory, Automata theory, Mathematical preliminaries Infinite Sets, Closures, Alphabets, Languages & Representation	**3**

2	Finite Automata and Regular Languages, Deterministic finite automata, Regular operations, Non-deterministic finite automata, Equivalence of DFA sand NFAs, Closure under the regular operations	**9**
3	Regular expressions, Equivalence of regular expressions and regular languages, The pumping lemma and non-regular languages	**5**
4	Context-Free Languages, Context-free grammars, Parse Trees & Ambiguity, Chomsky and Greibach Normal forms	**5**
5	Push Down Automata, Equivalence of PDA and CFG, Properties of context free languages, Determinism & Parsing DCFG, Top-down & Bottom-up Parsing	**9**
6	Turing Machine-Introduction, Recursive and Recursively Enumerable Language, Multi-tape Turing machines, Non-deterministic Turing machines, Universal Turing machines, Halting problem	**5**
7	Decidable and Undecidable Languages; Complexity Theory, The Complexity Class NP, NP Completeness and Reducibility NP complete problems	**6**
Total lectures		**42**

Suggested Text Book(s):

1. Elements of the Theory of Computation. Harry Lewis, Christos Papadimitriou, Second Edition, Pearson Education, 1998
2. Theory of computer Science: Automata, Language and Computation - KLP Mishra N Chandra Sekhran PHI, 3rd edn.

Suggested Reference Book(s):

1. Formal Languages and Automata- Peter Linz ,Narosa Pub. 4th edn.
2. M. Sipser, Introduction to the Theory of Computation, Thomson Asia, 1997.
3. J. E. Hopcroft, R. Motwani and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson, 2001.
4. D. C. Kozen, Automata and Computability, Springer-Verlag, 1997.

Other useful resource(s):

1. https://nptel.ac.in/courses/111103016/
2. Link to topics related to course:
   i. https://ocw.mit.edu/courses/mathematics/18-404j-theory-of-computation-fall-2006/
   ii. http://www.aduni.org/courses/theory/
   iii. http://www.cse.iitd.ernet.in/~sak/courses/toc/
   iv. http://cse.iitkgp.ac.in/~abhij/course/theory/FLAT/Spring13/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	2	3	3	2	1	3	2	**2.27**
CO-2	3	3	3	2	2	3	3	3	1	3	3	**2.64**
CO-3	3	3	3	2	2	3	2	3	2	2	3	**2.55**
CO-4	3	3	3	2	2	2	3	2	2	3	2	**2.46**
CO-5	3	3	3	2	2	2	2	2	1	2	2	**2.18**
Average	**3**	**2.8**	**2.8**	**2**	**2**	**2.6**	**2.6**	**2.4**	**1.4**	**2.6**	**2.4**	

Object Oriented Systems and Programming

COURSE CODE: 18B11CI311

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Programming for Problem Solving

Course Objectives

1. To strengthen their problem-solving ability by applying the characteristics of an object- oriented approach.
2. To apply the object-oriented programming concepts using C++.
3. To strengthen ability to design and represent solutions to problems using UML notations.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn the concepts of objects, classes, methods, constructors and destructors	Usage
CO-2	To learn the designing of complex classes: friend functions and static member functions, Inline functions, and constant member functions.	Usage
CO-3	To learn the concept of inheritance: single inheritance, multiple inheritance, multi-level inheritance, hierarchical inheritance and hybrid inheritance.	Usage
CO-4	To learn the concept of abstract classes, static and dynamic binding	Usage
CO-5	To learn the concepts of operator overloading and conversion function	Usage
CO-6	To learn file handling concepts. Writing and reading data from the file, reading and writing the objects into the file.	Usage
CO-7	To learn the concept of exception handling in C++.	Usage
CO-8	To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.	Familiarity

Course Contents:

Unit	Contents	Lectures required
1	Structured versus object-oriented programming, object-oriented paradigm. Defining objects, classes, data members, member functions, constructors and destructors in C++. Inline functions, friend functions, friend classes, constant member functions, and static members (static data and static member functions). Function overloading, operator overloading, and conversion functions. File handling: Opening and closing a file, writing into the file, reading from the file, searching and deleting froma file, updating a file.	**12**

2	Polymorphism and inheritance (single, multiple, multilevel, hierarchical, and hybrid) in C++. Abstract classes, static and dynamicbinding,virtualfunction,purevirtualfunction,andvirtual base class, virtual table, virtual pointer, and object slicing.	**8**
3	Function templates and class templates. Inheriting a template class from a template class, a template class from a non-template class, a non-template class from a template class.	**4**
4	Exception handling: Throwing and catching mechanisms, specifying an exception, re-throwing an exception, used-defined custom exceptions.	**4**
5	Advanced features in C++: Smart pointers, pointers to members, explicit and mutable keywords, namespaces, and run time type identification (RTTI)	**5**
6	Basic principles of Software engineering. System analysis, design, testing and debugging. Unified Modeling Language (Class Diagram, Use Case Diagram, State Diagram, Sequenced Diagram, Communication Diagram, Activity Diagram)	**9**
Total lectures		**42**

Suggested Text Book(s):

1. Lafore R., Object oriented programming in C++, WaiteGroup
2. Satzinger, Jackson, Burd, Object-Oriented Analysis &Design

ReferenceBook(s)

1. Stroustrap B., The C++ Programming Language, AddisonWesley
2. Bruce Eckel, Thinking in C++

Other useful resource(s):

1. Link to NPTEL coursecontents:
   - ttps://nptel.ac.in/courses/106/105/106105151/
2. Link to topics related tocourse:
   - ttps://www.tutorialspoint.com/cplusplus/
   - ttp://www.cplusplus.com/doc/tutorial

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO1	3	3	3	3	2	2	3	3	2	3	3	**2.73**
CO2	3	3	3	2	1	1	2	3	2	3	3	**2.36**
CO3	3	3	2	2	2	1	2	2	2	3	3	**2.27**
CO4	3	3	3	2	2	1	2	3	2	3	3	**2.45**
CO5	3	3	3	3	2	1	2	3	2	1	3	**2.36**
C06	3	3	3	3	2	3	2	3	2	2	3	**2.64**
CO7	3	3	2	2	2	3	3	2	2	3	3	**2.55**
CO8	3	3	3	3	3	2	3	3	2	1	3	**2.64**
Average	**3**	**2.89**	**2.78**	**2.44**	**2**	**1.89**	**2.44**	**2.78**	**2**	**2.44**	**2.89**	

Object-Oriented Systems and Programming Lab

COURSE CODE: 18B17CI371

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

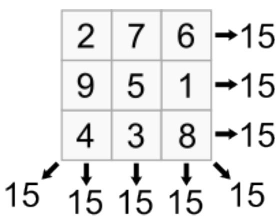
1. To use object modelling technique to analyse problem requirements, design a solution to the problem and then implement the solution in Object-Oriented Programming Language(s) or database.
2. To strengthen their problem-solving ability by applying the characteristics of an object-oriented approach.
3. To strengthen ability to design and represent solutions to problems using UML notations.
4. To introduce object-oriented concepts in C++ and Java.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn the concepts of Objects, Classes, Methods, Constructors and Destructors	Usage
CO-2	To learn the designing of complex classes: Friend Functions and Static member, functions, Inline functions, constant functions	Usage
CO-3	To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance and Hybrid Inheritance.	Usage
CO-4	To learn the concept of Abstract classes and interfaces	Usage
CO-5	To learn the concepts of Operator overloading and conversion function	Usage
CO-6	To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file.	Usage
CO-7	To learn the Exception Handling: try --catch and finally block, making user-defined exceptions.	Usage
CO-8	To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.	Familiarity

List of Experiments

S. No	Description	Hours
1	Define a class in Java to represent a bank account of 10 customers having following members:Datamembers: Nameofthedepositor	**4**

	Accountnumber Typeofaccount Balanceamountin theaccount Memberfunctions(Defineallthememberfunctionsinsidetheclass): Toassigninitialvalues To depositanamount Towithdrawanamountafter checkingthe balance Todisplaynameandbalance	
2	Createa Javaprogramthatcomputesthe distance anobjectwillfallinEarth'sgravity.	**4**
3	Write a class in Java for users having profile Name, Password, and Access Rights. Access rightstake on values X, R, W, and ALL. It is possible to have default values for Password and Accessrights which are the first three letters of the Name and ALL respectively. Users can change theirpassword andaccessrights.	**4**
4	Write a Java program to calculate pay for hourly employees of an organization. The employeesget paid time and a half for any hours over 40 that they work in a single week. For example, if anemployeeworks45hours,theyget5hoursofovertime,at1.5timesthei rbasepay.However,theorganizationalso requiresthatanemployeenotworkmorethan60 hoursin aweek. Note:*Donottrytowritetheentireprograminonego.Itismucheasiertow riteasmallpieceandtest it, then write another small piece and test it. For example, start by writing just a skeleton ofyour method and your main program. Then add the code to do the normal salary computation,without any special rules. Then add each additional rule, one at a time. You should test yourprogramwith simpletestinputsto checkthatyou handleeachcase.*	**4**
5	Buildanelectronicrentallibrarysystemhavingtwolibraries.Eachlibrar yhasdifferentaddressesand book collections offers many books to rent. Customers can print the list of available books,borrow, andreturn books. Note: *Some methods will need to be static methods, and some need to be instance methods. Becareful when comparing Strings objects. Use string1.equals(string2) for comparing the contentsofstring1 and string2.*	**4**
6	Write a program to implement a magic square of order n is an arrangement of n × n numbers,usually distinct integers, in a square, such that the n numbers in all rows, all columns, and bothdiagonalssumto thesameconstant.  Figure:Amagic squareoforder 3,Source:Wikipedia)	**4**

7	WriteaprograminJavathatdemonstrateshandlingofexceptionsininheritancetree.Forexample,createabaseclass'Father'andsubclasscalled'Son'whichextendsthebaseclass.InFatherclass,implement a constructor which takes the age and throws the exception WrongAge () where theinputage<0.InSonclass,implementaconstructorthatusesbothfatherandson'sageandthrowsanexception ifson'sageis>=father'sage.	**4**
8	DefineanabstractclassShapeinJava,withtwoabstractfunctions:getData()anddisplayArea().Define two classes Circle and Rectangle inside a new file that inherit the class Shape andimplement the abstract functions in their own way. Illustrate the concept of dynamic binding inthisprogram.	**4**
9	Define a class Person in the package MyPackage. The class has data members as: name, age,address and member functions to enter the data through keyboard and display them. Make use ofoverloaded constructors in the class. Now, import the above class and inherit the class Employeefromthis.Thesub-classshouldhaveoverloaded functionsandalsocalltothebaseclassconstructors.In themain class,defineadatabaseof 5 objectsand displaythem.	**4**
10	Find the source code for the Vector class in the Java source code library that comes with all Javadistributions.CopythiscodeandmakeaspecialversioncalledintVectorthatholdsonlyintegers.Consider what it would take to make a special version of Vector for all the primitive types. Nowconsider what happens if you want to make a linked list class that works with all the primitivetypes. If parameterized types are ever implemented in Java, they will provide a way to do thiswork foryouautomatically.	**4**
11	Create a class MyString in Java containing a String object so that you initialize in the constructorwithargument.AddatoString()methodandamethodconcatenate()thatappendsaStringobjectto your internal string. Implement clone () in MyString. Create two static methods that each takea MyString x handle as an argument and call x.concatenate("test"), but in the second method callclone()first.Testthetwomethods andshow thedifferenceeffects.	**4**
12	To implement database connectivity in Java using the JDBC API by creating a simple librarymanagement system. The program should display a menu with options to add, retrieve, update,and delete books by performing CRUD (Create, Read, Update, Delete) operations on a MySQLdatabase.	**6**
13	Youaretaskedwithdevelopingaminie-commerce systemthatallowsforthefollowingoperations: Adding new customers. Addingnew products. Placingordersfor customers. Retrievingorderdetailsforacustomer,includingproductinformation andtotalorderamount. Implementingtransactionstoensureorderplacementisatomic.	**6**

	Note:*Thisexerciseshouldinvolvemultipletableswithrelationships,performingcomplexquerieswithjoins and useoftransactionsto ensuredata integrity.*	
Total Lab hours		**56**

SuggestedBooks/Resources:

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evalution	20
2	End Sem. Evalution	20
3.	Attendance	15
4.	Lab Assessment	45
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO1	3	3	3	3	2	2	3	3	2	3	3	2.73
CO2	3	3	3	2	1	1	2	3	2	3	3	2.36
CO3	3	3	2	2	2	1	2	2	2	3	3	2.27
CO4	3	3	3	2	2	1	2	3	2	3	3	2.45
CO5	3	3	3	3	2	1	2	3	2	1	3	2.36
C06	3	3	3	3	2	1	2	3	2	2	3	2.45
CO7	3	3	2	2	2	3	3	2	2	3	3	2.55
CO8	3	3	3	3	3	2	3	3	2	1	3	2.64
Average	3	2.89	2.78	2.44	2	1.67	2.44	2.78	2	2.44	2.89	

Database Management Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Introduction to Computer Programming, Discrete Mathematics, Data Structures

Course Objectives:

1. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
2. To understand and use data manipulation language to query, update, and manage a database.
3. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
4. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the characteristics, architecture of database approach, its components, different data models and the examples of their usage.	Understand
CO-2	For a given specification of the requirement, design the databases using E-R method and normalization.	Apply
CO-3	For a given query write relational algebra expressions for that query and optimize the developed expressions.	Apply
CO-4	Determine the functional dependency between two or more attributes, compute the closure of a set of attributes, evaluate a proposed decomposition	Evaluate
CO-5	Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.	Apply
CO-6	Familiarize with the security in databases and gaining familiarity with other popular databases used in the industry	Understand

Course Contents:

Unit	Contents	Lectures required
1	Fundamentals of Database and Database Management System, Physical Level of Data Storage, Structure of Relational Database, Table, Attributes, Records	**3**

2	Introduction to SQL, Data Types in SQL, Operations on Single Table like Create, Insert, Delete, Update, Alter, etc., SQL Queries on Single Table using Select Statement with or without Where and Group By Clause, etc., Overview of No SQL Databases	**4**
3	Database Design and ER Model, Entity type, Attributes, Relation types, Notations, Constraints, Extended ER Features	**5**
4	Relational Model and Structured Query Language, Data Definition and Data Manipulation, Relational Algebra	**7**
5	Procedural Language, PL/SQL, Stored Procedures, Functions, Cursors, Triggers	**5**
6	Functional Dependency & Normalization, Data Dependencies, 2NF, 3NF, BCNF, Building Normalized Databases	**8**
7	Transaction Management, ACID properties of Transactions, Need of Concurrency, Concurrency Control Techniques, Deadlocks, Database Recovery System, Database Security and Authorization	**10**
Total lectures		**42**

Suggested Text Book(s):

1. R. Elmasri, S. B. Navathe , "Fundamentals of Database Systems", Pearson Education.
2. H. F. Korth, A. Silberschatz, S. Sudarshan, “Database system concepts”, McGraw-Hill Ellis.
3. C. J. Date, “An Introduction to Database Systems”, Pearson Education.
4. Bayross, “SQL, PL/SQL – The Programming Language of Oracle” BPB Publications.

Suggested Reference Book(s):

1. J. D. Ullman, J Windon, “A First Course in Database Systems”, Pearson, Education.
2. R. T. Watson, “Data Management: Databases and Organization”, Wiley.

Other useful resource(s):

1. Fundamentals of Database Systems, IIT Kanpur | https://nptel.ac.in/courses/106104135

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
CO-6	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.6**	**2.3**	**2.3**	**2.2**	**2.0**	**1.3**	**1**	**1**	**1.6**	**1.8**	**1.3**	

Database Management Systems Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Basics of Programming Language

Course Objectives:

1. To develop the ability to design, implement and manipulate databases.
2. To build database management systems.
3. To apply DBMS concepts in real life applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Design and implement a database schema using Structured Query Language	Apply
CO2	Design different views of tables for different users and to apply embedded and nested queries	Apply
CO3	Understand, analyze and apply common SQL statements including DDL, DML, DCL statements to perform different operations	Evaluate, Apply
CO4	Develop application programs using PL/SQL	Apply

List of Experiments

S. No	Description	Hours
1	To implement Data Definition language and Data Manipulation Language commands: Create database/table, alter, drop, truncate, Insert, Select, Update, Delete etc.	2
2	To implement Constraints as a part of Data Definition language: Primary key, Foreign Key, Check, Unique, Null, Not null, Default, Enable Constraints, Disable Constraints, Drop Constraints	2
3	To practice in Built Functions: Date functions, numerical functions, character functions, conversion functions, group functions, count functions etc.	2
4	To practice group by, having clause and special operators such as between, like, in etc.	2
5	To practice Joins and Set Operators Union, Intersect, Minus, Except	2
6	To implement Views from Single and Multiple Tables	2
7	To implement the ER Diagram modelled for different problems	2
8	To implement and practice PL/SQL control structure if, if then else, else if, nested if, for and while loop	2
9	To implement PL/SQL Procedures, Functions, Cursors and Triggers	6

10	To implement Transaction Processing Commands: Commit, Rollback, Save Point and Time Stamps	**2**
11	To implement Data Control Language Commands: Grant and Revoke	**2**
12	Implementation of database connectivity using a programming language (e.g., Python, PHP) to perform basic database operations.	**2**
	Total Lab hours	**28**

Suggested/Resources:

1. I. Bayross, “SQL, PL/SQL – The Programming Language of Oracle” BPB Publications.
2. R. Elmasri, S. B. Navathe , "Fundamentals of Database Systems", Pearson Education.
3. H. F. Korth, A. Silberschatz, S. Sudarshan, “Database system concepts”, McGraw-Hill Ellis.

Other useful resource(s):

1. NPTEL - Fundamentals of Database Systems, IIT Kanpur | https://nptel.ac.in/courses/106104135
2. Coursera - Database Structures and Management with MySQL | https://www.coursera.org/learn/database-structures-and-management-with-mysql

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Unix Programming Lab

Course Code: XXXXXXXX
Course Credit: 2
Core/Elective: Core
L-T-P: 1-0-2

Course Objective:

1. Introduces basic understanding of UNIX OS, UNIX commands and File and directory system.
2. Implement basic shell commands, I/O redirection, and background jobs.
3. To make student learn fundamentals of shell script and shell programming.
4. Emphases are on system-level programming concepts using Unix system calls, networking, and archiving utilities.

Course Outcomes:

S. No.	Course Outcomes	Attainment Level
CO1	Familiarity with command-line navigation and file manipulation and ability to combine Unix commands with shell control flow.	Familiarity
CO2	Understand inter-process data streaming in Unix, understand processhierarchiesandsystemcallinterfacesandshellparsing, job control, and processmanagement.	Assessment
CO3	Understand file descriptors and buffered vs unbuffered I/0 and learn how Unix represents and accesses directories.	Apply
CO4	Develop shell scripts using control structures, loops, and command-line arguments to automate tasks.	Apply
CO5	Apply system-level programming concepts using Unix system calls, directory handling, networking, and archiving utilities.	Apply

List of Content:

S. No.	Description	(L+P) Hours
1	Introduction to Unix Environment Objective: Learn basic Unix commands and directory navigation. Tasks: Use pwd, ls, cd, mkdir, rmdir, cp, mv, cat, touch.	1+2
2	User and Group Management Objective: Understand multi-user environment basics. Tasks: Use useradd, passwd, groupadd, whoami, id. File Permissions and Ownership Objective: Understand file permissions and access control. Tasks: Use chmod, chown, chgrp, umask to modifypermissions.	1+2
3	File and Text Operations Objective:Learntoview,concatenate,andmanipulatetextfiles. Tasks:Usemore,less,head,tail,sort,uniq,cut,paste.	1+2
4	Searching and Pattern Matching Objective: Explore file searching and text filtering. Tasks: Use grep, egrep, find, locate for file and pattern search.	1+2
5	Input/Output Redirection and Piping Objective: Combine commands and manage data flow. "asks: Use >,>>,<, I, tee for redirection and pipelines.	1+2
6	Process and Job Control Objective: Manage background and foreground processes. Tasks: Use ps, *top,* jobs, kill, bg, fg, nice.	1+2

7	System Information Commands Objective: Retrieve system status and user information. Tasks: Use who, w, uname, uptime, df, du, free, date, cal.	1+2
8	Conditional Execution in Shell Objective:Useif,else,andelifinscripts. Tasks:Checkfileexistence,testnumericconditions(even/odd). Objective: Apply for, while, and untilloops. Tasks: Print numbers, calculate sums, generate multiplication tables.	1+2
9	Command Line Arguments in Shell Objective: Pass and use arguments in shell scripts. Tasks: Perform arithmetic operations using positional parameters.	1+2
10	String and File Handling in Shell Objective: Manipulate strings and files programmatically. Tasks: Countwords/lines,reversestrings,readfilesline-by-line	1+2
11	Shell scripting Objective: perform useful application Tasks: Data summarization (std, mean, correlation etc), Report generation	
12	File Handling and System call Objective: Understand file descriptor, buffer v/s unbuffered I/O Tasks: Open(), read(), write(), lseek(), close() Objective: Director traversal utility Tasks: opendir(), readdir(), and state() system calls.	1+2
13	Archiving and Compression Objective: Manage backups and compressed files Tasks: Use tar, gzip, gunzip,zip, unzip.	1+2
14	Basic Networking inUnix Objective: Use networking utilities for connectivity and monitoring. Tasks: Use ping, ifconfig, netstat, scp, ssh.	1+2
	Total Hours	14+28

Suggested Textbook(s):

1. BrianW.KernighanandRobPike,TheUNIXProgramming Environment,PrenticeHallIndia. (AvailableintheLRCandas aneBookonthestudentresourceportal)
2. Sumitabha Das, UNIX: Concepts and Applications, Tata McGraw Hill, LatestEdition.
3. W. Richard Stevens, Advanced Programming in the Unix Environment,Pearson
4. W. Richard Stevens, Unix Network Programming, Pearson, 3rd edition

Suggested Reference Book(s):

1. Stephen Prata, Advanced UNIX: A Programmer's Guide, The Wait Group, SamsPublishing.
2. Jack Tackett Jr. and David Gunter, Using Linux: The Most Complete Reference, 3rd Edition.Publishing.
3. A Ramasatish, Unix Programming,SCITECH

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20

2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO 4	PO 5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	Averag e
CO-1	3	2	2	2	3	2	1	1	1	1	1	1.72
CO-2	2	3	3	3	3	2	1	2	2	2	3	2.36
CO-3	3	2	2	1	2	3	2	1	1	1	2	1.81
CO-4	3	3	3	3	3	2	2	2	2	1	2	2.36
CO-5	2	1	1	1	1	1	1	1	1	1	1	1.09
Averag e	2.6	2.2	2.2	2	2.4	2	1.4	1.4	1.4	1.2	1.8	

ObjectOriented Programming using Java

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Basic knowledge of programming fundamentals using languages like C or Python.

Course Objectives:

1. To introduce students to the Java programming environment and develop the ability to construct reusable and modular classes using fundamental OOP concepts.
2. To provide practical experience in implementing object-oriented relationships such as inheritance, association, aggregation, and composition in Java.
3. To develop skills in abstraction and modularization using abstract classes, interfaces, packages, and memory management techniques like garbage collection.
4. To strengthen students' ability to write reliable Java programs using exception handling, string manipulation, and the Java Collections Framework.
5. To enable students to apply advanced programming concepts such as multithreading, generics, and reflection, along with developing GUI-based Java applets.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Apply object-oriented concepts to design and implement reusable and modular Java classes and programs.	Apply
CO2	Demonstrate implementation of relationships such as Inheritance, Association, Aggregation, and Composition in Java.	Understand
CO3	Develop Java applications using abstract classes, interfaces, packages, and effective memory management practices.	Apply
CO4	Build robust Java programs with proper exception handling and use of collection frameworks for data manipulation.	Evaluate
CO5	Develop Java programs that demonstrate concurrency using multithreading, utilize generics for type-safe programming.	Evaluate
CO6	Design interactive Java applets and demonstrate the ability to embed and parameterize them in HTML pages.	Apply
CO7	Analyze string handling in Java and implement programs using the String class and its methods effectively.	Evaluate

List of Experiments

S. No	Description	Hours
1	Introduction to Java and its features, Function of JVM, and JDK, Class and primitive data types in java.	**2**

2	Implementation of class in Java involving tokens, variables, methods, operators, control statements, array, static variables and methods, reference data types, variables and methods, constructors.	**2**
3	Implementation of Inheritance and other relationships like Association, Aggregation and Composition in Java.	**4**
4	Implementation of abstract class and methods, interface, packages, and garbage collection in Java.	**4**
5	Writing programs in Java to handle strings.	**2**
6	Java programs involving exception handling, exception propagation, and user defined checked & unchecked exceptions.	**2**
7	Writing Java programs involving Collections framework, Collection interfaces, Collection classes, Iterators, Comparators, User defined classes in Collections, etc.	**4**
8	Implementing Multithreading in Java and Java programs involving Generic classes and methods, Wild cards, Metadata & Reflection	**4**
9.	Writing Java programs to implement Applet fundamentals, applet class, HTML Applet tag, passing parameters to applets.	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. Herbert Schildt, *Java: The Complete Reference*, 12th Edition, McGraw Hill, 2024.
2. Kathy Sierra and Bert Bates, *Head First Java,* 3rd Edition, O'Reilly Media, 2022.
3. Paul Deitel and Harvey Deitel, *Java: How to Program,* 11th Edition, Pearson, 2018.
4. Joshua Bloch, *Effective Java,* 3rd Edition, Addison-Wesley, 2018.
5. Link to topics related to the course:
   i. NPTEL – Programming in Java (IIT Kharagpur) | https://nptel.ac.in/courses/106105191
   ii. Oracle Java Tutorials (Official Documentation) | https://docs.oracle.com/javase/tutorial/
   iii. GeeksforGeeks – Java Programming Language | https://www.geeksforgeeks.org/java/
   iv. JavaTpoint – Java Tutorial | https://www.javatpoint.com/java-tutorial
   v. Coursera – Object Oriented Programming in Java (Duke University) | https://www.coursera.org/learn/object-oriented-java

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	3	3	3	1	1	1	1	1	3	**1.90**
CO-2	2	2	3	3	3	2	1	2	2	1	3	**2.36**
CO-3	2	2	3	3	3	2	1	2	2	1	3	**2.18**
CO-4	2	2	3	3	3	2	1	2	2	1	3	**2.18**
CO-5	2	2	3	3	3	2	1	1	1	1	3	**2.0**

CO-6	2	2	3	3	3	2	1	2	2	1	2	**2.18**
CO-7	2	2	3	3	3	1	1	1	1	1	2	**1.82**
Average	**2.0**	**2.0**	**3.0**	**3.0**	**3.0**	**1.71**	**1**	**1.57**	**1.57**	**1**	**2.71**	

Competitive Programming -I

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Software Development Fundamentals – I & II

Course Objectives:

1. To develop analytical problem-solving skills.
2. To strengthen the programming skills in C/C++.
3. To prepare students for national and international programming contests
4. To enhance technical coding ability for placement and interviews.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO1	Apply problem-solving strategies to develop efficient solutions for algorithmic problems.	Apply
CO2	Participate confidently in online programming contests.	Apply
CO3	Implement and use C/C++ programming concepts.	Apply
CO4	Prepare for technical interviews by solving coding-related problems with realistic constraints.	Create, Evaluate

List of Experiments

S. No	**Description**	**Hours**
1	**Introduction to Competitive Programming:** Online Platforms (HackerRank, GeeksforGeeks, CodeStudio, LeetCode), Test Cases, Demo of Passing Test Cases to the program. Submission of Codes.	**2**
2	**C++ OOP Concepts:**Class, Object, Constructor, Inheritance, Polymorphism, Function Overloading	**2**
3	**Control Structures & Number Problems -** Nested loops, Decision making	**2**
4	**Arrays (1D & 2D):** Rotation, rearrangement, subarrays, Searching techniques	**2**
5	**Strings: C**-strings & string in C++, Palindromes, pattern matching, anagrams, longest substring	**4**
6	**Functions & Recursion:** User-defined functions, Recursive functions	**4**
7	**Pointers & Dynamic Memory:** Pointers, Memory allocation (malloc, calloc), Pointer to array and functions	**4**

8	**Structures, Unions, and Files:** Structures, nested structures, File handling (text files)	**4**
9	**Pseudo Code & Logic -** Output Prediction, Error Finding, Loops, Strings, Arrays, Pointers, Functions, Recursion	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. "Cracking the Coding Interview", by Gayle Laakmann McDowell, CareerCup; 6th edition.
2. "Test Your C++ Skills", by Y. P. Kanetkar, BPB Publications, 2011.
3. "Stroustrup: The C++ Programming Language (4th Edition)," by Bjarne Stroustrup, Addison-Wesley; 4th edition.
4. "Programming interviews exposed : secrets to landing your next job" , by John Mongan, Giguère E., and N. Kindler, John Wiley & Sons, 2013.

Online Practice Platforms:

1. GeeksforGeeks C/C++ Placement Practice – Topic-wise questions with solutions.
2. InterviewBit / LeetCode / HackerRank

Evaluation Scheme:

1	Mid Sem. Evaluation (50-question test)	20 Marks
2	End Sem. Evaluation (Online Assessment test)	20 Marks
3	Attendance	10 Marks
4	Assignments (Also includes weekly contests on Online Platforms)	20 Marks
5	Daily Lab Assessments (including Mock tests, Timely submission of codes, Passing of test cases, etc.)	30 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	-	-	-	-	-	3	-	-	-	**2.6**
CO-2	3	3	2	3	2	-	-	3	-	-	-	**2.7**
CO-3	3	3	2	2	-	-	-	3	-	-	-	**2.6**
CO-4	3	2	3	2	2	-	-	3	-	-	-	**2.5**
Average	**3**	**2.5**	**2.3**	**2**	**2**	-	-	**3**	-	-	-	

Digital Systems and Computer Organisation

COURSE CODE: 25B11CI411

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

1. To design and simulate digital circuits using multiplexers, decoders, and comparators.
2. Hands-on implementation of flip-flops, counters, and shift registers enhances their practical understanding.
3. Programming ALUs, memory units, and I/O interfaces builds a strong foundation in computer architecture.
4. Prepare students for real-world applications in digital and embedded systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain Number systems, Boolean algebra, Basics of Computer Systems, Instruction sets and I/O organizations.	Understanding
CO-2	Implement digital logic techniques for designing Combinational circuits, and Flip flops.	Apply
CO-3	Analyze various Sequential circuits and examine the behavior of Memory hierarchy, Cache mapping techniques, and Virtual memory systems	Analyze
CO-4	Evaluate the performance trade-offs of different CPU design approaches, including hardwired vs. microprogrammed Control.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	Number systems: Binary, Octal, Hexadecimal, Boolean algebra: logic gates, K-maps; SOP/POS.	**5**
2	Combinational Circuits: Half Adder, Half Subtractor, Full Adder, Full Subtractor, Multiplexer, Demultiplexer, Encoder, Decoder, Comparator and Code Converter	**7**
3	Sequential Circuits: Flip Flops: SR, JK, Master Slave JK, T And D; Excitation Tables, Conversion of Flip-Flops Synchronous and Asynchronous Counters, Design of Counters Using Flip- Flops, Analysis of Sequential Circuits, State Diagram Design, Registers,	**10**

	Shift Registers, Counters Using Shift Registers	
4	Basics of Computer Systems and Instruction Set Architecture (ISA): Introduction to Computer Organization, Types of Instructions: Data Movement, Arithmetic, Control Flow, Addressing Modes: Immediate, Direct, Indirect, Register, Indexed, Instruction Formats (RISC vs. CISC Concepts)	**7**
5	CPU Design and Control Unit: CPU Components: ALU, Registers, Control Unit, Hardwired vs. Microprogrammed Control, Instruction Cycle: Fetch-Decode-Execute, Design of ALU	**6**
6	Memory & Input- Output Organization: Memory Hierarchy: Cache, Main, Secondary, Cache Memory: Mapping Techniques (Direct, Associative, Set-Associative), Virtual Memory, Paging, TLB, Memory Access Time and Performance, I/O Interfaces and Data Transfer Techniques, Programmed I/O, Interrupt-Driven I/O, DMA.	**7**
Total lectures		**42**

Suggested Text Book(s):

1. M. M. Mano and M. D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, 6th edition, Pearson, 2021
2. Ramesh S. Gaonkar, "Microprocessor Architecture, Programming and Applications with the 8085," 6th edition, Penram International Publishing, 2013.

Suggested Reference Book(s):

1. D. A. Patterson and J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface (MIPS Edition), 5th edition, Morgan Kaufmann, 2013
2. W. Stallings, Computer Organization and Architecture: Designing for Performance, 11th Global edition, Pearson, 2022
3. Thomas L. Floyd, "Digital Fundamentals, Global Edition," 11th edition, Pearson Education, April 2025.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/117105080
2. Link to NPTEL course contents: https://nptel.ac.in/courses/117106114

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)- 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.7**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.7**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.1**
Average	**2**	**2.5**	**2.5**	**2.5**	**2.5**	**1**	**1**	**1**	**2**	**2.3**	**1.5**	

Digital Systems and Computer Organization Lab

COURSE CODE:25B17CI471

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-1

Pre-requisite: Computer Fundamentals

Course Objectives:

1. To understand and implement basic combinational and sequential digital circuits.
2. To analyze and design digital systems using logic gates and flip-flops.
3. To gain hands-on experience with hardware description and simulation tools.
4. To study and implement basic components of computer organization such as ALU, registers, and memory.
5. To develop problem-solving skills for designing and testing digital and computer-based systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Understand and analyze the functionality of basic logic gates and verify their truth tables.	Analyze
CO2	Apply the concepts of logic gates to Design and simulate combinational circuits such as adders, subtractors, decoders, multiplexers, and code converters.	Apply
CO3	Implement and analyze sequential circuits including SR latch, D flip-flop, and JK flip-flop.	Apply
CO4	Create a 2-bit and 4-bit Arithmetic Logic Unit (ALU) using MATLAB for performing basic operations.	Apply

List of Experiments

S. No	Description	Hours
1	Introduction to MATLAB/ Digital Integrated Circuits.	4
2	To verify the truth tables of basic logic gates (AND, OR and NOT) and their realization using universal logic gates (NAND and NOR) using Digital Integrated Circuits.	2
3	To design a half adder and a full adder circuits using Digital Integrated Circuits and verify their functionality.	2

4	To design a half subtractor and a full subtractor using Digital Integrated Circuits and verify their functionality.	2
5	To design a 2-to-4 decoder and a 3-to-8 decoder using Digital Integrated Circuits and verify their functionality.	2
6	To design a 2-to-1, a 4-to-1, and a 8-to-1 multiplexers using Digital Integrated Circuits and verify their functionality.	2
7	To design a binary-to-gray, and a gray-to-binary code converters using Digital Integrated Circuits and verify their functionality.	2
8	To realize an SR Latch using Digital Integrated Circuits and verify its functionality.	2
9	To realize (i) a D flip flop using Digital Integrated Circuits and verify its functionality. (ii) aJK flip flop using Digital Integrated Circuits and verify its functionality.	4
10	To design a 2-bit and a 4-bit ALUs using Digital Integrated Circuits (or MATLAB) and verify their functionality.	2
11	**Mini Project**: Students will learn about Combinational and Sequential logic circuits and design them using open source software/MATLAB. Additionally, students in group sizes of two-three will realize various applications of Digital Systems employing these circuits.	4
Total Lab hours		28

Suggested/Resources:

1. Salivahanan, S., and S. Arivazhagan. Digital circuits and design.Vikas publishing house PVT Limited.
   Fifth edition (March 2018)
2. M. Morris Mano, Computer System Architecture, Pearson, 2006.
3. M. M. Mano, "Digital Logic and Computer Design", Pearson Education India.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	3	2	2	3	1	1	3	1	2	2.18

CO-2	3	3	3	3	3	3	1	1	1	1	2	2.18
CO-3	3	3	3	3	3	2	1	1	1	1	1	2
CO-4	3	3	3	3	3	3	2	2	1	1	1	2.45
Average	3	3	3	2.75	2.75	2.75	1.25	1.25	1.75	1	1.50	

Design and Analysis of Algorithms

COURSE CODE: 25B11CI412

COURSE CREDITS: 3

CORE/ELECTIVE: Core

L-T-P: 3-1-0

Pre-requisite: Data structure and algorithm

Course Objectives:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the concepts of algorithms and their analysis.	Understand
CO-2	Apply algorithmic design techniques to solve fundamental computational problems.	Apply
CO-3	Analyze a given problem to identify an appropriate data structure and algorithm design strategy.	Analyze
CO-4	Evaluate the efficiency and suitability of different algorithmic approaches for solving a given problem.	Evaluate

Course Contents:

Unit	Subtitle of the Module	Contents	No. of Lectures
1.	Introduction	Analyse the performance of algorithms; Asymptotic analysis and notations- big O, big omega, big theta, etc. Growth of Functions; Recurrence and Solving Recurrences using different approaches; Empirical analysis of sorting and searching algorithms	9
2.	Design Technique: Divide and Conquer	Fundamentals of Divide and Conquer (D&C) approach, designing the D&C based algorithms for binary search, quick sort, merge sort, matrix multiplication, closest pair, etc.	5
3.	Design Technique: Backtracking Algorithms	Fundamentals of backtracking algorithms and designing of backtracking based solutions for N-queen, Rat in a maze, etc. problems and designing backtracking based solutions for M-coloring problem, Hamiltonian Cycle detection, Travelling salesman problem, Network flow, etc	6

4.	Design Technique: Greedy Algorithms	Fundamentals of greedy algorithms and designing solutions for graph problems like Shortest Path, Minimum Spanning Trees (Prim's and Kruskal algorithms), Coloring, Covering, etc., fractional Knapsack, Coinage problem, Bin packing, Job scheduling, Text compression using Huffman coding & Shannon-Fanon coding, etc.	8
5.	Dynamic Programming	Fundamentals of Dynamic Programming (DP) and designing DP based solutions for 0/1 Knapsack, Shortest path, Coinage problem, Matrix Chain Multiplication, Longest common subsequence, Longest increasing sequence, String editing, etc.	7
6.	String Algorithms	String matching algorithms using Naïve String Matching, Finite Automata Matcher, Rabin Karp matching algorithm, Knuth Morris Pratt, etc; Review of string based data structures: Suffix Tree, Suffix Array and Tries.	6
7.	Tractable and Non-Tractable Problems	Efficiency and Tractability, P, NP, NP-Complete, NP- Hard problems	1
		Total lectures	**42**

Suggested Text Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, MIT Press, 4th Edition, 2022
2. Steven Skiena,The Algorithm Design Manual, Springer; 3rd Edition, 2020
3. Knuth, The art of Computer Programming Volume 1, Fundamental Algorithms, Addison-Wesley Professional; 3rd Edition,1997
4. Knuth, The art of Computer Programming Volume 3, Sorting and Searching, Addison-Wesley Professional; 2nd Edition,1998
5. Horowitz, Sahni and Rajasekaran, Fundamentals of Computer Algorithms, Computer Science Press, 2008
6. Sedgewick, Algorithms in C, 4th Edition. Addison Wesley, 2016
7. Alfred V. Aho, J.E. Hopcroft, Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley Series in Computer Science and Information Processing, 1983

Suggested Reference Books:

1. Tim Roughgarden, Algorithms Illuminated: Part 1: The Basics, Sound like yourself Publishing, September
   27, 2017
2. Tim Roughgarden, Algorithms Illuminated: Part 2: Graph Algorithms and Data Structures,Soundlike yourself Publishing, First Edition, 2018.
3. Tim Roughgarden, Algorithms Illuminated:Part3: Greedy Algorithms and Dynamic Programming, Sound likey ourself Publishing, First Edition, 2019.
4. Weiss, Data Structures and Algorithm Analysis in C++, 4th Edition, Pearson, 2014

Other useful resource(s):

1. https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/
2. Link to NPTEL course contents:
   https://nptel.ac.in/courses/106106131
   http://www.digimat.in/nptel/courses/video/106101060/L01.html

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)- 10 Attendance - 5

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	1.72
CO-2	2	3	3	3	3	1	1	1	2	2	1	2
CO-3	2	2	2	2	3	1	1	1	2	2	1	1.72
CO-4	1	2	2	1	3	2	1	1	1	1	1	1.45
Average	1.75	2.25	2.25	2	2.75	1.25	1	1	1.75	1.75	1.25	

Design and Analysis of Algorithms Lab

COURSE CODE: 25B17CI472

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Data Structures and Algorithms

Course Objectives:

4. Analyze asymptotic performance and correctness of algorithms
5. Apply divide and conquer to design efficient algorithms
6. Use backtracking to explore combinatorial search spaces
7. Design greedy algorithms and prove optimality
8. Formulate dynamic programming solutions for optimal substructure problems
9. Master string algorithms for pattern matching and text processing

Course Outcomes

S No	Course Outcomes	Level of Attainment
CO1	Understand and define appropriate data structure to a given problem	Apply
CO2	Understand complexity using asymptotic and experimental analysis for various algorithmic design techniques.	Apply
CO3	Apply and build various algorithms and design techniques to solve the given problem.	Apply
CO4	Formulate, elaborate, and design an efficient solution to a given problem using appropriate data structure and algorithm design technique.	Apply, Analyze

List of Experiments (To be Implemented using C/C++, or Java)

S No	Description	Hours
1	Implement and compare the runtime performance of the following sorting algorithms on identical input data and visualize how their execution times scale with array size. • Insertion Sort • Selection Sort • Merge Sort • Quick Sort (random pivot) • Heap Sort	**4**
2	Implement the base-case logic of Strassen's Algorithm to compute the product of two 2×2 matrices using a specific set of recursive formulas that prioritize reducing the number of multiplications.	**2**
3	You are given a graph with *N* vertices**.** The goal is to assign one of *M* colors to each vertex. Constraints: • No two adjacent vertices can have the same color. • If such a coloring exists, print it; otherwise, report that no solution exists.	**2**
4	You are given a set of *N* cities and the distances between each pair of cities. The salesman must start from a city, visit every other city exactly once**,** and return to the starting city. Find the shortest possible route that satisfies these conditions.	**2**

5	Place N queens on an $N \times N$ chessboard. Constraint: No two queens can attack each other. That means no two queens share the same row, column, or diagonal. Backtracking Approach Recursive placement: • Place a queen in a row. • Check if the position is safe (no conflicts with previously placed queens). • If safe, recurse to the next row. • If not safe, try the next column. • If no column works, backtrack to the previous row. Base case: • If all queens are placed successfully, print the board.	**4**
6	Write a program to compute optimal parenthesization and minimum multiplication cost for a chain of matrices.(Matrix Chain Multiplication)	**2**
7	Implement Prim's algorithm to compute the Minimum Spanning Tree (MST) of a weighted undirected graph and print the MST weight and edges.	**2**
8	Implement Huffman coding to compress and decompress a given text string. Build the Huffman tree from character frequencies, generate prefix codes, encode the input into a bitstring of 0s and 1s, then decode that bitstring back to the original text. ***Input*** • A single line containing the text to encode. The text may include spaces and printable ASCII characters. ***Output*** • First line: **k** the number of distinct symbols. • Next **k** lines: each symbol and its code as symbol code using readable labels for whitespace (for example SPACE for space). • Next line: Encoded followed by the encoded bitstring (sequence of 0 and 1). • Final line: Decoded followed by the decoded text. ***Constraints*** • Use a min-heap priority queue to build the tree. • Codes must be prefix free. • Handle empty input and single-symbol input correctly. • A plain string of 0/1 bits is acceptable for the encoded output; bit packing is optional.	**4**
9	Implement a C/C++/Java program that finds all occurrences of a pattern P in a text T using the Rabin–Karp algorithm. The program should compute a rolling hash for each length-m window in T, compare it to the pattern hash, and verify matches by direct comparison when hashes match.	**2**
10	Mini Project: Designing an efficient solution to a given problem using appropriate data structure and algorithm design technique.	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, MIT Press, 4th Edition, 2022
2. Steven Skiena,The Algorithm Design Manual, Springer; 3rd Edition, 2020
3. Knuth, The art of Computer Programming Volume 1, Fundamental Algorithms, Addison-Wesley Professional; 3rd Edition,1997
4. Knuth, The art of Computer Programming Volume 3, Sorting and Searching, Addison-Wesley Professional; 2nd Edition,1998
5. Horowitz, Sahni and Rajasekaran, Fundamentals of Computer Algorithms, Computer Science Press, 2008
6. Sedgewick, Algorithms in C, 4th Edition. Addison Wesley, 2016
7. Alfred V. Aho, J.E. Hopcroft, Jeffrey D. Ullman, Data Structures and Algorithms, Addison-Wesley Series in Computer Science and Information Processing, 1983

Recommended Reading material:

1. Tim Roughgarden, Algorithms Illuminated: Part 1: The Basics, Sound like yourself Publishing, September 27, 2017
2. Tim Roughgarden, Algorithms Illuminated: Part 2: Graph Algorithms and Data Structures, Soundlike yourself Publishing, First Edition, 2018.
3. Tim Roughgarden, Algorithms Illuminated:cPart3: Greedy Algorithms and Dynamic Programming, Sound like yourself Publishing, First Edition, 2019.
4. Weiss, Data Structures and Algorithm Analysis in C++, 4th Edition**,** Pearson, 2014

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Project based learning:

Students in a group of 4-5 will have to develop a mini project based on appropriate data structures and algorithm design techniques. The students can opt any real-world application where these algorithms can be applied. The students have to implement the mini project using C/C++/Java language. Project development and its presentation will enhance coding skills, knowledge and employability of the students in IT sector.

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	2	1	1	1	2	**1.82**
CO-2	3	2	3	3	3	2	1	2	2	1	2	**2.18**
CO-3	2	2	2	2	2	2	1	2	2	2	2	**1.91**
CO-4	3	3	3	3	3	2	1	2	2	3	2	**2.45**
Average	**2.75**	**2.25**	**2.5**	**2.5**	**2.75**	**1.75**	**1.25**	**1.75**	**1.75**	**1.75**	**2**	

Artificial Intelligence and Machine Learning

COURSE CODE: 25B11CI413

COURSE CREDITS: 3

CORE/ELECTIVE: Core

L-T-P: 3-0-0

Pre-requisite: Python Programming, Data Structures, Linear Algebra

Course Objectives:

1. Understand the fundamental concepts of Artificial Intelligence, including problem-solving strategies and intelligent agents.
2. Apply intelligent search techniques to formulate and solve real-world computational problems.
3. Implement supervised, unsupervised, and semi-supervised learning algorithms for data-driven problem solving.
4. Apply machine learning algorithms to classification problems, analyzing their performance and limitations.
5. Analyze and compare deep learning architectures and evaluate their suitability for real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the basics of artificial intelligence, problem solving strategies, and machine learning methods	Understand
CO-2	Apply intelligent searching techniques to solve a given problem	Understand
CO-3	Apply supervised, unsupervised or semi-supervised learning algorithms for a given problem	Apply
CO-4	Apply the machine learning algorithms for classification problems	Apply
CO-5	Analyze the different deep learning architectures for real-world applications.	Apply

Course Contents:

Unit	Contents	Lectures required
1	Introduction to AI, Problems of AI - Search, reasoning, planning, learning, perception, AI technique, Tic – Tac – Toe Problem as a state-space search problem. Intelligent Agents: Definition, properties, and rationality, Agents & Environment, Nature of Environment, Structure of Agents, Goal-based agents, Utility-based agents, learning agents, Identify PEAS (Performance, Environment, Actuators, Sensors) for: Vacuum cleaner agent, Self-driving car	**6**
2	Problem solving agents, searching for solutions; uniform search strategies: breadth first search, depth first search. Heuristic search strategies Greedy best -first search, A* search, AO* search.	**9**

3	Fundamentals of Machine learning, Types of Machine Learning: Supervised, unsupervised, reinforcement, Machine perception - feature extraction - classification, clustering, linear and logistic regression.	**9**
4	Concept of ANN (Artificial Neural Network): Perceptron and backpropagation neural network - k-nearest neighbor rule. Support vector machine: Decision trees: and random forest.	**9**
5	Introduction to Deep learning, Convolutional neural networks, CNN Architectures LeNet, AlexNet, GooleNet, VGG Net, ResNet: Comparative analysis	**9**
Total lectures		**42**

Suggested Text Book(s):

1. R. O. Duda, E. Hart, and D.G. Stork, “Pattern Classification”, Second Edition, John Wiley & Sons, Singapore, 2012.
2. Francois Chollet, “Deep Learning with Python”, Manning Publications, Shelter Island, New York, 2018
3. Satish Kumar, “Neural Networks A Classroom Approach”, McGraw Hill Education (India) Pvt. Ltd, 2010
4. S. Russell and P. Norvig, “Artificial Intelligence: A Modern Approach”, Prentice Hall, Third Edition, 2015.

Suggested Reference Book(s):

1. EthemAlpaydin, “Introduction to Machine Learning”, 3rd Edition, MIT Press, 2014.
2. C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2006.
3. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012
4. Elaine Rich, Kevin Knight, & Shivashankar B Nair, “Artificial Intelligence”, McGraw Hill, 3rd ed.,2017.
5. Patterson, “Introduction to Artificial Intelligence & Expert Systems”, Pearson, 1st ed. 2015.

Other useful resource(s):

1.Link to NPTEL course contents:https://nptel.ac.in/courses/106102220

2.https://nptel.ac.in/courses/106106139

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 5 Quizzes(2)-15 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Artificial Intelligence & Machine Learning Lab

COURSE CODE: 25B17CI473

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Python Programming Language

Course Objectives:

1. To familiarize students with the practical implementation of fundamental Artificial Intelligence and Machine Learning algorithms using standard programming tools and libraries.
2. To enable students to preprocess, clean, and prepare real-world datasets for effective machine learning model development.
3. To develop hands-on proficiency in implementing supervised and unsupervised learning techniques and evaluating their performance using appropriate metrics.
4. To train students in applying classification, regression, clustering, and dimensionality reduction techniques to solve standard engineering problems.
5. To inculcate good laboratory practices such as result analysis, model comparison, documentation, and interpretation of experimental outcomes.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO1	Implement basic Artificial Intelligence and Machine Learning algorithms using standard programming tools and libraries.	Apply
CO2	Preprocess and prepare datasets by applying data cleaning, transformation, and feature selection techniques.	Apply
CO3	Analyze and evaluate the performance of supervised and unsupervised learning models using appropriate metrics.	Analyze
CO4	Design and implement machine learning solutions for classification, regression, and clustering problems.	Create
CO5	Interpret experimental results, compare models, and document outcomes following standard laboratory practices.	Evaluate

List of Experiments

S. No	**Description**	**Hours**
1	**Introduction to Python and Scientific Libraries** Study and use of NumPy, Pandas, Matplotlib, and Scikit-learn for ML experiments.	**4**
2	**Data Preprocessing and Exploratory Data Analysis** Handling missing values, normalization, encoding categorical data, and basic visualization.	**4**

3	**Implementation of Linear and Multiple Linear Regression** Predicting continuous outcomes and evaluating model performance.	**2**
4	**Implementation of Logistic Regression** Binary classification and performance evaluation.	**2**
5	**Implementation of k-Nearest Neighbour (k-NN) Algorithm** Distance metrics and model tuning.	**2**
6	**Implementation of Decision Tree Algorithm** Tree construction, pruning, and result interpretation.	**2**
7	**Implementation of Support Vector Machine (SVM)** Linear and non-linear classification using kernels.	**2**
8	**Implementation of k-Means Clustering** Unsupervised learning and cluster analysis.	**2**
9	**Mini Project / Case Study using Machine Learning** End-to-end problem: data preprocessing, model building, evaluation, and documentation.	**2**
10	**Breadth First Search (BFS), Depth First Search (DFS), and A* search algorithms** To implement and compare Breadth First Search (BFS), Depth First Search (DFS), and A* search algorithms, and to study the effect of heuristics on search efficiency and optimality.	**2**
11	**Implementation of Principal Component Analysis (PCA)** Dimensionality reduction and variance analysis.	**2**
12	**Implementation of Artificial Neural Network (ANN)** Feed-forward neural network using standard libraries.	**2**
	Total Lab hours	**28**

Suggested/Resources:

6. M. Lutz, Programming Python, 4th ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2010.
7. Tom Mitchell, "Machine Learning", McGraw Hill, 1997, ISBN 0070428077.
8. Sebastian Raschka, "Python Machine Learning", Packt Publishing Ltd.
9. Andreas C. Müller, Sarah Guido, "Introduction to Machine Learning with Python", O'Reilly Media, Inc.
10. SunilaGollapudi, "Practical Machine Learning", Packt Publishing Ltd
11. Wes McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", O'Reilly Media, Inc.
12. A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019
13. K. P. Murphy, Machine Learning: A Probabilistic Perspective. Cambridge, MA, USA: MIT Press, 2012.
14. S. Weidman, Deep Learning from Scratch: Building with Python from First Principles. Sebastopol, CA, USA: O'Reilly Media, 2019.
15. Willi Richert, "Building Machine Learning Systems with Python", Packt Publishing Ltd.

16. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	-	3	-	-	-	-	-	-	**2.33**
CO-2	2	3	2	2	3	-	-	-	-	-	-	**2.40**
CO-3	2	3	2	3	2	-	-	-	-	-	-	**2.40**
CO-4	2	3	3	2	3	-	-	-	-	-	-	**2.60**
CO-5	1	2	2	2	2	-	-	2	-	3	-	**1.833**
Average	**2.00**	**2.60**	**2.20**	**2.25**	**2.60**	**-**	**-**	**2.00**		**3.00**	**-**	

Software Engineering

COURSE CODE: 25B11CI414

COURSE CREDITS: 3

CORE/ELECTIVE: Core

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

1. Provide students with a systematic understanding of software engineering principles and software process models used in the development of large-scale software systems.
2. Enable students to identify, analyze, and document functional and non-functional requirements and prepare a well-structured Software Requirement Specification (SRS).
3. Develop the ability to apply software design principles such as modularity, cohesion, and coupling, and to model software systems using UML diagrams.
4. Equip students with knowledge of software construction practices, coding standards, metrics, and software testing techniques to ensure software quality.
5. Encourage students to evaluate and improve software systems based on quality attributes, performance metrics, and testing outcomes, preparing them for professional software development roles.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand software engineering principles and various process models for project development.	Understand
CO-2	Outline the functional and non-functional requirements of a software project for creating Software Requirement Specifications (SRS).	Understand
CO-3	Apply software design principles (modularity, cohesion, coupling) and create UML models.	Apply
CO-4	Analyze the performance of software system using different testing methodologies.	Analyze
CO-5	Evaluate the software in terms of general software quality attributes, code structure and try to optimize the overall software system.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	***Introduction to Software Engineering:*** Introduction to Software Engineering Principles, Software Process Models (Build and Fix Model, Waterfall Model, Incremental Process Model, Evolutionary- Prototype and Spiral Models, Agile Models (tools study). Project Planning, Project Scheduling: Network Diagram, Gant	**7**

	Chart, CPM and PERT.	
2	***Requirement Engineering:*** Types of Requirement, Requirement Elicitation, Analysis, Specification, SRS, Requirement Verification and Validation.	**4**
3	***Software Design:*** Use Case Diagram, State Diagram, Activity Diagram, Class Diagram, Sequence Diagram, Collaboration Diagram, Deployment Diagram, Component Diagram and Package Diagram, Design Modularity: Coupling Cohesion.	**7**
4	***Software Construction:*** Coding Standards and Guidelines, Code Checklist, Code Reviews, Code Refactoring, Code Optimization. Design Pattern, Modern Programming Environments (Code Search, Programming using Library Components and their APIs), Program Comprehension; Program Correctness, Defensive Programming.	**8**
5	***Software Metrics:*** Size-Oriented Metric, Function-Oriented Metric, Halstead's Software Metric, Information Flow Metric, Object-Oriented Metric, Class-Oriented Metric, COCOMO Model.	**7**
6	***Software Testing:*** White-Box Testing, Basis Path Testing, Control Structure Testing: Condition Testing, Data Flow Testing, Loop Testing, Black-Box Testing: Equivalence Class Partitioning, Boundary Value Analysis, Decision Table Testing, Cause Effect Graphing, Mutation Testing and Regression Testing, Formal Methods.	**9**
	Total lectures	**42**

Project-Based Learning: Each student works on a different case study in Assignments. They utilize the concepts taught in lecture and develop project in a group of 3-4.

The course emphasized on the skill development for employability in software industry by engaging students on Software Development methodologies. Various activities are carried out to enhance the student's software development skills. Some of them are study of various software process models and their applicability, progress tracking, size estimation techniques, software testing strategies, etc.

Suggested Text Book(s):

1. Roger S. Pressman, "Software Engineering: A Practitioner's Approach", Fifth Edition-TMH International.
2. Sommerville, "Software Engineering", Seventh Edition - Addison Wesley.

Suggested Reference Book(s):

1. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, Reading, Massachusetts, May 2005
2. Richard Thayer, "Software Engineering Project Management", Second Edition -Wiley-IEEE Computer Society Press.
3. B. Bezier, "Software Testing Techniques", Second Edition- International Thomson Computer

Press.

4. Pankaj Jalote, "An Integrated Approach to Software Engineering" Third addition, Springer Press

Other useful resource(s):

1.Link to NPTEL course contents:https://nptel.ac.in/courses/106105182

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO / PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2.00**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1.0**	**1.0**	**1.8**	**2.0**	**1.4**	

Competitive Programming -II

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Software Development Fundamentals, Data Structures

Course Objectives:

1. To develop analytical problem-solving skills.
2. To strengthen the programming skills.
3. To Design and implement optimized solutions using appropriate data structures.
4. To prepare students for national and international programming contests
5. To enhance technical coding ability for placement and interviews.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Implement and use programming concepts.	Apply
CO2	Analyze the time and space complexity of different algorithms.	Analyze
CO3	Implement non-linear data structures such as trees and graphs.	Apply
CO4	Design and develop optimized and real-time solutions using appropriate algorithms and data structures.	Create
CO5	Prepare for technical interviews by solving coding-related problems with realistic constraints.	Create, Evaluate
CO6	Participate confidently in online programming contests.	Apply

List of Experiments

S. No	Description	Hours
1	**Linked Lists:** Reversal, merging, cycle detection, Intersection, palindrome check	**4**
2	**Stacks & Queues:** Balanced parentheses, Next greater/smaller element, Implement stack/queue using array/linked list, Queue using stacks and vice versa, infix-postfix conversion	**4**
3	**Sorting Algorithms:** Bubble Sort, Selection Sort, Insertion Sort	**4**
4	**Trees:** Binary Tree, BST basics, Tree traversals (Inorder, Preorder, Postorder), Lowest Common Ancestor (LCA)	**6**
5	**Hashing:** Frequency count, duplicates, subarray sum	**4**

6	**Graphs:** BFS and DFS, Detect cycle in a graph, Shortest Path (Dijkstra, Bellman-Ford), Topological Sort, Minimum Spanning Tree (Prim's, Kruskal's), Union-Find / Disjoint Set.	**6**
Total Lab hours		**28**

Suggested/Resources:

1. "Cracking the Coding Interview", by Gayle Laakmann McDowell, CareerCup; 6th edition.
2. "Competitive Programming 3", Steven Halim, Lulu.
3. "Programming interviews exposed : secrets to landing your next job" , by John Mongan, Giguère E., and N. Kindler, John Wiley & Sons, 2013.
4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, third edition. MIT Press, 2009.

Online Practice Platforms:

1. GeeksforGeeks C/C++ Placement Practice – Topic-wise questions with solutions.
2. InterviewBit / LeetCode / HackerRank

Evaluation Scheme:

1	Mid Sem. Evaluation (50-question test)	20 Marks
2	End Sem. Evaluation (Through Online Assessment test)	20 Marks
3	Attendance	10 Marks
4	Assignments (Including weekly contests on Online Platforms)	20 Marks
5	Daily Lab Assessments (including Mock tests, Timely submission of codes, Passing of test cases)	30 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	-	1	-	-	3	-	-	-	**2**
CO-2	3	2	2	1	2	-	-	3	-	-	-	**2.2**
CO-3	3	3	2	2	-	-	-	3	-	-	-	**2.6**
CO-4	3	3	3	2	2	-	-	3	-	-	-	**2.7**
CO-5	3	2	3	2	2	-	-	3	-	-	-	**2.5**
CO-6	3	3	2	3	2	-	-	3	-	-	-	**2.7**
Average	**3**	**2.5**	**2.2**	**2**	**1.8**	-	-	**3**	-	-	-	

Operating Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: C/C++

Course Objectives:

1. To learn the mechanisms of OS to handle processes and threads and their communication.
2. To learn the mechanisms involved in memory management in contemporary OS
3. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.
4. To know the components and management aspects of concurrency management

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To appreciate the role of an operating system.	Understand
CO-2	Create processes and threads.	Create, Evaluate
CO-3	Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, and Response Time.	Create, Evaluate
CO-4	For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time	Evaluate
CO-5	Design and implement file management system.	Create, Evaluate
CO-6	For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.	Apply

Course Contents:

Unit	Contents	Lectures required
1	Introduction and Historical context of Operating Systems, Operating Structure and Architecture	3
2	Process: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.	3
3	Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling Algorithms: Pre-emptive and Non preemptive, FCFS, SJF, Priority,	7

	Round Robin, Multiple processor scheduling	
4	Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson's Solution, Lamport's Bakery Algorithm, The Producer\ Consumer Problem, Semaphores, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc. Monitors	**10**
5	Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.	**4**
6	Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.	**6**
7	Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Not recently used (NRU) and Least Recently used (LRU).	**4**
8	Case studies of OS: Windows, Linux, IBM, Tizen Operating System, Distributed O.S: Introduction to distributed operating systems, synchronization and deadlock in distributed systems	**2**
Total lectures		**42**

Suggested Text Book(s):

1. "Operating System Concepts" 9th Edition by Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. "Operating Systems: Internals and Design Principles" 9th Edition, William Stallings, Pearson.

Suggested Reference Book(s):

1. "Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. "Operating Systems: A Modern Perspective" 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106108101/
2. Link to topics related to course:
   a. https://nptel.ac.in/courses/106106144/
   b. https://www.class-central.com/course/udacity-introduction-to-operatingsystems-3419

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	3	2	2	3	2	2	3	2	3	**2.5**
CO-2	3	3	3	2	3	2	2	2	3	2	3	**2.5**
CO-3	3	3	3	2	2	3	2	3	3	2	3	**2.6**
CO-4	3	3	3	2	3	3	2	3	3	2	3	**2.7**
CO-5	3	3	3	2	3	3	2	3	3	2	3	**2.7**
CO-6	3	3	3	2	3	3	2	2	3	2	3	**2.6**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2.5**	**3**	**2**	**3**	

Operating Systems Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: C/C++

Course Objectives:

1. To write programs in Linux environment using system calls.
2. To implement the scheduling algorithms.
3. To implement page replacement algorithms
4. To implement file allocation methods.
5. To understand and implement ipc mechanism using named and unnamed pipes.
6. To develop solutions for synchronization problems using semaphores.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO1	Understand Various Unix Commands	Understand
CO2	Develop programs to create different types of processes using pthread library under Linux environment.	Apply
CO3	Develop programs to implement resource management task like CPU scheduling algorithms, deadlock handling.	Apply
CO4	Develop programs to implement and test various synchronization techniques like semaphores, binary semaphore and monitors via different classical test suites.	Apply
CO5	Design and analyse various disk-scheduling algorithms, memory management schemes, file management systems.	Analyze

List of Experiments

S. No	**Description**	**Hours**
1	Learning Unix Commands (file commands, directory commands, symbolic links, terminal commands, help commands, information commands, useful cshell symbols, permissions and file storage (Unix), permissions and file storage (andrew), processes, printing Environment, customizing networking, x-applications unix filters)	**6**
2	Develop programs to create different types of processes under Linux environment. Develop programs to create multitasking threads using pthread library under Linux environment. Develop programs to implement inter-process communication	**4**

3	Develop programs to implement resource management task like CPU scheduling algorithms (First Come First Serve, Shortest Job First, Round Robin, Priority Scheduling, Multi level Queue), deadlock handling (Prevention, Avoidance and Detection)	**8**
4	Develop programs to implement and test various synchronization techniques like semaphores, binary semaphore and monitors via different classical test suites.	**4**
5	Design and analyze various disk-scheduling algorithms. Develop programs to implement memory management schemes. Design, implement and assess file management systems (file organization and file directories).	**6**
Total Lab hours		**28**

Suggested/Resources:

1. Charles Crowley "Operating System A Design Approach"TMH.
2. Andrew S. Tanenbaum "Operating Systems Design and Implementation", Third Edition, Prentice Hall Publications2006
3. A.S. Tanenbaum, "Modern Operating Systems", 2nd edition, Prentice Hall India.
4. A.Silberschatz, P.Galvin, G. Gagne, "Operating systems concepts" Willey international company (sixth edition)
5. Gary Nutt, "Operating Systems – A modern perspective", Pearson Education
6. David Solomon and Mark Russinovich ," Inside Microsoft Windows 2000", Third Edition, Micorosoft Press
7. D. M. Dhamdhere, " Systems Programming and Operating systems" TMH, 2nd revised edition.2006
8. ACM/IEEE transactions on operating systems concepts.
9. www.vmware.com

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.7**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.2**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.1**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.2**
CO-5	3	2	3	3	3	3	2	1	2	2	2	**2.3**
Average	**3**	**2.6**	**2.6**	**2.8**	**3**	**2**	**1.2**	**1.6**	**1.8**	**1.2**	**2**	

Computer Networks

COURSE CODE: XXXXXX

COURSE CREDITS: 4

CORE/ELECTIVE: CORE

L-T-P: 3-1-0

Pre-requisite: Basics of Computers and C Language

Course Objectives:

1. The course introduces the concepts and fundamental design principles of modern computer networking, focusing on the Internet architecture and protocols.
2. The course introduces the concepts of data and computer communications, computer network introduction and its applications in our real life.
3. Reference models such as OSI and TCP/IP and its way toward the physical layer concepts, data link layer and its protocols, multiple access protocols.
4. Provide students Network layer and its different routing protocols, the concepts/design of IP addressing.
5. To provide students Transport layer and its protocols such as TCP, UDP and SCTP to application layer and its protocol such as HTTP, FTP, SMTP and DNS.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn the basic concepts and terminology in computer networks and understand Physical Layer.	Understand
CO-2	To learn about Data Link Layer, Protocols, Addressing and issues.	Evaluate
CO-3	To learn about Network Layer, IPv4 and IPv6 Addressing, subnetting, supernetting, routing mechanisms and Protocols.	Evaluate
CO-4	To learn about Transport Layer, Addressing, protocols, Congestion Control	Evaluate
CO-5	To learn about the Application Layer, Protocols and related issues	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	Overview of Computer Networks, Network Hardware, Network Software, Networks Topologies, Layering and Protocols: OSI, TCP/IP Reference Models, ARPANET, Internet	**4**
2	Physical Layer: Guided Transmission Media: Twisted Pairs, Fiber Optics, etc., Errors in Transmission: Attenuation, Noise, Repeaters, Encoding (NRZ, NRZI, etc.), Wireless Transmission	**6**

3	Data link layer: Design issues, framing, Error detection (Parity, CRC) and correction, Elementary data link protocols: simplex protocol, stop and wait protocols for error-free channel and noisy channel, Sliding Window Protocols, Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching	**9**
4	Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Broadcast, Multicast, Distance vector, Inter-domain routing, etc., Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet, Internet Protocol, IPv6, ARP, DHCP, ICMP, Network Address Translation	**9**
5	Transport Layer: Transport Services, Elements of Transport protocols, Connection management: establishment and termination, TCP and UDP protocols, flow and congestion control, timers, retransmission, TCP extensions, etc.	**8**
6	Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video, Session, Presentation, and Application Layers Examples: DNS, SMTP, IMAP, etc.	**6**
Total lectures		**42**

Suggested Text Book(s):

1. B.A. Forouzan “Data Communication and Networking”,McGraw Hill Edication, 5th Edition, 2013.
2. Andrew S. Tanenbaum, “Computer Networks” 4th Edition PHI
3. James F. Kurose, Keith W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet” 3rd Edition Pearson Education

Suggested Reference Book(s):

1. UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998, ISBN 0-13-490012-X.
2. Arnold Robbins, “UNIX in a Nutshell”, O‟Reilly 4th Edition
3. David I. Schawartz, “Introduction to UNIX”, Prentice Hall, Second Edition
4. BEHROUZ a. Forouzan and Richard F. Gilberg, “UNIX and Shell Programming: A Textbook”
5. NS Simulator for Beginner‟s, Lecture notes Univ. de Los Andes, France.
6. Angela Orebaugh, Gilbert Ramirez,Josh Burke, Larry Pesce, Joshua Wright, Greg Morris, “Wireshark & Ethereal Network Protocol Analyzer Toolkit”, Syngress Publishing, Inc.

Other useful resource(s):

Link to topics related to course:

1. https://www.coursera.org/specializations/networking
2. https://nptel.ac.in/courses/106105080/
3. https://swayam.gov.in/course/4066-computer-networks

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	3	2	2	3	3	2	2	3	2	**2.5**
CO-2	3	3	3	2	3	2	2	2	2	3	2	**2.5**
CO-3	3	3	3	2	3	3	2	2	3	3	1	**2.6**
CO-4	3	3	3	2	2	3	2	2	3	3	1	**2.5**
CO-5	3	3	3	2	3	2	2	2	3	3	1	**2.5**
Average	**3**	**3**	**3**	**2**	**2.6**	**2.6**	**2.2**	**2**	**2.6**	**3**	**1.4**	

Computer Networks Lab

COURSE CODE: XXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: Core

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. To implement important computer networking protocols in a high-level programming language.
2. To understand the working principle of various communication protocols.
3. To analyze the various routing algorithms.
4. To know the concept of data transfer between nodes.
5. To become acquainted with socket programming and some of the important computer networking tools.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To understand and implement the working concepts of Networking and inter-networking devices.	Apply
CO-2	To implement different network topology designs and their characteristics.	Apply
CO-3	To implement different routing algorithms.	Apply
CO-4	To understand and implement the concepts of client–server interaction using connection oriented and connectionless protocols.	Apply
CO-5	To analyze different application layer protocols using network analysis tools.	Analyze

List of Experiments:

Sr. No.	Description	Hours
1	To study Network Topology Design & Simulation using Cisco Packet Tracer.	2
2	To study VLAN Configuration and Inter-VLAN Routing.	2
3	To study different graph traversal algorithms in context of computer network	2
4	Finding shortest path between any two nodes in a computer network using Dijkstra's and Prim's shortest path algorithm.	2
5	Finding articulation point in an undirected graph.	2
6	Static and Dynamic Routing (RIP/OSPF) using PacketTracer.	2
7	Implementation of CRC generator and checker algorithm in C/C++/Java.	2

8	Implementation of Hamming code algorithm in C/C++/Java.	2
9	Implementing client-server programusing TCP/UDP sockets.	2
10	HTTP/DNS Protocol Analysis using Wireshark.	2
11	Implementation of Stop-and-Wait protocol in C / C++ / Java in a client-server environment using sockets	2
12	Implementation of encryption algorithm converting plaintext to ciphertext using C / C++ / Java	2
13	Design and implement Traffic Shaping Algorithms: a) Leaky Bucket b) Token Bucket	2
14	Implementing Packet Capture and Analysis using Wireshark (ICMP, ARP, TCP)	2
	Total Lab Hours	**28**

Suggested Resources:

1. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition, Pearson Education.
2. Andrew S. Tanenbaum, Computer Networks, 4th Edition, Prentice Hall of India.
3. W. Richard Stevens, UNIX Network Programming, Volume 1: Networking APIs: Sockets and XTI, 2nd Edition, Prentice Hall, 1998. ISBN: 0-13-490012-X.
4. Arnold Robbins, UNIX in a Nutshell, 4th Edition, O'Reilly Media.
5. David I. Schwartz, Introduction to UNIX, 2nd Edition, Prentice Hall.
6. Behrouz A. Forouzan and Richard F. Gilberg, UNIX and Shell Programming: A Textbook.
7. Lecture Notes, NS Simulator for Beginners, University of Los Andes.
8. Links to topics related to course:
   i. https://www.coursera.org/specializations/networking-basics
   ii. https://nptel.ac.in/courses/106105080/
   iii. https://swayam.gov.in/course/4066-computer-networks

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-project etc.)	60
	Total	**100**

Course Outcomes (Cos) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO1	3	3	3	3	3	3	1	3	3	3	1	**2.64**
CO2	2	2	2	2	2	1	1	1	2	2	3	**1.81**
CO3	3	3	3	3	3	1	1	3	3	1	3	**2.45**
CO4	2	2	2	2	2	2	1	1	2	2	2	**1.82**
CO5	2	2	2	2	2	2	1	2	2	2	2	**1.91**
Average	**2.4**	**2.4**	**2.4**	**2.4**	**2.4**	**1.8**	**1**	**2**	**2.4**	**2**	**2.2**	

Full Stack Web Development Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: HTML, SQL

Course Objectives:

1. To introduce students to the Django web framework and full-stack development.
2. To train students in developing modular web applications with dynamic content.
3. To understand template inheritance and efficient front-end integration.
4. To implement database models and perform CRUD operations.
5. To introduce AJAX for asynchronous operations in Django.

Course Outcomes:

Sr. No.	Course Outcomes	Level of Attainment
CO1	Understand Django framework and set up full-stack web environments.	Understand
CO2	Build dynamic web applications with views, templates, and models.	Create
CO3	Use Django admin and perform model-based operations.	Apply
CO4	Implement AJAX and file generation (CSV/PDF) in web applications.	Create, Evaluate

List of Experiments

Sr. No.	Description	Hours
1	Introduction to Python, Django and VS Code. Setup virtual environment, create Django project	**2**
2	Create a Django app and understand file structure, settings, and run server	**2**
3	Create a Django view to display current server time with offsets (4 hours ahead and before)	**2**
4	Develop a Django app that displays: Unordered list of fruits	**1**
5	Extend previous lab to display: Ordered list of students selected for an event	**1**
6	Develop a multipage Django website with navigation and layout using template inheritance	**2**

7	Create student and course models, with ManyToMany relationship through Enrolment	**2**
8	Create student registration form to enroll students in selected courses	**2**
9	Extend registration app to include admin site customization and user login functionality	**2**
10	Add field-level validation for forms and display error messages using Django messages framework	**2**
11	Implement CSV file generation for student-course registration data	**2**
12	Implement PDF generation for course or student details	**2**
13	Build AJAX-powered search interface to display enrolled courses by searched student	**2**
14	Final Project Lab: Integrate all features into a complete Django application	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. Vincent, William S. "Django for beginners: build websites with Python & Django." 2018.
2. Dauzon, Samuel, AidasBendoraitis, and Arun Ravindran. “Django: web development with Python.”Packt Publishing Ltd, 2016.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO1	3	2	2	1	3	1	1	2	2	2	2	**1.91**
CO2	3	3	3	2	3	2	1	2	3	2	2	**2.36**
CO3	3	2	3	2	3	1	1	2	3	3	2	**2.27**
CO4	2	2	2	2	3	2	1	2	3	2	3	**2.18**
Average	**2.75**	**2.25**	**2.5**	**1.75**	**3**	**1.5**	**1**	**2**	**2.75**	**2.25**	**2.25**	

Competitive Programming -III

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Data Structures, Design and Analysis of Algorithms

Course Objectives:

1. To develop analytical problem-solving skills.
2. To develop a strong understanding of the core principles of recursion, memoization, and dynamic programming for solving complex computational problems.
3. To enable students to identify and decompose problems exhibiting overlapping subproblems and optimal substructure.
4. To train students in designing and implementing both top-down and bottom-up approaches using tabulation and memoization techniques
5. To enhance technical coding ability for placement and interviews.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Implement and use C++/Java programming concepts.	Apply
CO2	Analyze the time and space complexity of dynamic programming solutions.	Analyze
CO3	Develop and implement bottom-up and top-down DP approaches for common algorithmic problems.	Apply
CO4	Solve classical DP problems such as LCS, LIS, Knapsack, Matrix Chain Multiplication, and Partitioning.	Apply
CO5	Solve real-world placement coding questions involving dynamic programming on platforms like LeetCode, GFG, and Codeforces.	Evaluate
CO6	Participate confidently in online programming contests.	Apply

List of Experiments

S. No	Description	Hours
1	**Sorting Algorithms:** Quick Sort, Merge Sort	**2**
2	**Recursion & Backtracking:** N-Queens, Rat in a Maze, Subset/Permutation generation	**4**
3	**Greedy Algorithms:** Activity selection, coin change (greedy)	**4**
4	**Dynamic Programming:** Climbing Stairs, Min Cost Climbing Stairs, House Robber Longest Increasing Subsequence, 0/1 Knapsack, Subset Sum, Coin Change, Longest Increasing Subsequence (LIS), Longest Common Subsequence (LCS), Matrix chain multiplication, Maximum	**10**

	Path in a Matrix, Egg Dropping Problem, Diameter of Tree, Maximum Path Sum, Wildcard Matching, DP on subsets, strings, grids,	
5	**Bit Manipulation (Basic):** Count set bits, Check power of 2, XOR tricks, Subset generation using bits	**4**
6	**Sliding Window:** Longest substring without repeating characters, Max sum of subarray of size k	**4**
7	**Two Pointers:** Pair with a given sum, Sorting-based problems	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. "Cracking the Coding Interview", by Gayle Laakmann McDowell, CareerCup; 6th edition.
2. "Elements of programming interviews: the insiders' guide", Adnan Aziz, T.-H. Lee, and Amit Prakash, Createspace Independent Pub; 2nd edition
3. "Dynamic Programming for Coding Interviews", Meenakshi and K. Rawat, Notion Press, 2017.
4. "Competitive Programming 3", Steven Halim, Lulu.

Online Practice Platforms:

1. GeeksforGeeks C/C++ Placement Practice – Topic-wise questions with solutions.
2. InterviewBit / LeetCode / HackerRank

Evaluation Scheme:

1	Mid Sem. Evaluation (50-question test)	20 Marks
2	End Sem. Evaluation (Online Assessment test)	20 Marks
3	Attendance	10 Marks
4	Assignments (Including weekly contests on Online Platforms)	20 Marks
5	Daily Lab Assessments (including Mock tests, Timely submission of codes, Passing of test cases)	30 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	-	1	-	-	3	-	-	-	**2**
CO-2	3	2	2	1	2	-	-	3	-	-	-	**2.2**
CO-3	3	3	3	2	1	-	-	3	-	-	-	**2.5**
CO-4	3	3	2	3	1	-	-	3	-	-	-	**2.5**
CO-5	3	3	3	2	2	-	-	3	-	-	-	**2.7**
CO-6	3	3	2	3	2	-	-	3	-	-	-	**2.7**
Average	**3**	**2.7**	**2.2**	**2.2**	**1.5**	**-**	**-**	**3**	**-**	**-**	**-**	

Web Technology

COURSE CODE: XXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: Core

L-T-P: 3-0-0

Pre-requisite: HTML, Javascipt, MySQL,Java Programming

Course Objectives:

1. To understand the concepts of WWW including Browser and HTTP Protocol and various HTML Tags.
2. To style websites using modern CSS3 layout models and responsive techniques.
3. To understand and use the JavaScript concepts to develop the Dynamic Web Pages.
4. To develop the Server-Side Scripting with PHP to generate the Web Pages dynamically using the Database Connectivity.
5. To learn and understand basics of Servlets and JSP.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To understand the concepts of world wide web (WWW) including browser and HTTP Protocol and various HTML Tags and use them to develop the user-friendly web pages	Understand
CO-2	To style the web pages using cascading style sheets (CSS) and various CSS rules.	Understand
CO-3	To master JavaScript fundamentals for programming interactivity and creating dynamic web pages.	Apply
CO-4	To develop Server-Side Scripting with PHP to generate the Web Pages dynamically that uses the database connectivity.	Apply
CO-5	To develop the modern web pages using JSP and learning Servlets with different layouts.	Apply

Course Contents:

Unit	Contents	Lectures required
1	Introduction to world wide web (WWW), Web concepts: The Client-Server Architecture, Browsers and HTTP Protocol, Introduction to website and web pages. Introduction to HTML: HTML Common tags- Block Level and Inline Elements, Lists, Links, Tables, Images, HTML Forms, other special HTML tags.	**6**

2	Cascading Style sheets, CSS Properties; CSS Syntax, Inline, Internal and External CSS, CSS Selectors – Element, ID and Class selector, Text, Font, Background, Border, Lists and Table properties, etc. Java Script: Introduction to Java Script, JavaScript syntax, including variable declaration (let, const), control structures (loops, conditions), and functions, pop-up boxes in Java Script, Objects in Java Script, Dynamic HTML with Java Script, DOM Manipulation & Event Handling.	**10**
3	Introduction to PHP: Basics of PHP, Functions, Error Handling, Interaction between PHP and MySQL, Database using Forms, Using PHP to manipulate and Retrieve Data in MySQL JDBC: Data Base, Database Schema, A Brief Overview of The JDBC Process, JDBC Driver Types, Creating, Inserting, Updating and Deleting Data In Database Tables, Result, Set, Metadata	**10**
4	Web Servers and Servlets: Tomcat web server, Introduction to Servlets: Servlets, the advantage of Servlets over "Traditional" CGI, Basic Servlet Structure, Simple Servlet, Generating Plain Text, Compiling and Installing the Servlet, Invoking the Servlet, Lifecycle of a Servlet.	**8**
5	Introduction to JSP: Introduction - Advantages of JSP - Developing first JSP - Components of JSP - Reading Request Information - Retrieving the Data Posted from a HTML File to a JSP File - JSP Sessions - Cookies - Disabling Sessions	**8**
Total lectures		**42**

Suggested Text Book(s):

3. Jon Duckett, HTML & CSS: Design and Build Websites (Wiley)
4. Jennifer Niederst Robbins, Learning Web Design (5th Edition), O'Reilly Media.

Other useful resource(s):

1. https://www.youtube.com/watch?v=QEtWL4lWlL4
2. https://www.youtube.com/watch?v=uUhOEj4z8Fo
3. https://www.youtube.com/watch?v=eR9jf-0tS78
4. http://www.digimat.in/nptel/courses/video/106105191/L01.html
5. https://www.youtube.com/watch?v=ussgXhJ-cp0

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Class Performance - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Web Technology Lab

COURSE CODE: XXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: HTML, Javascript, MySQL, Java Programming

Course Objectives:

1. To learn the concepts of WWW and creating website structure using HTML Tags.
2. To style websites using modern CSS layout models and responsive techniques.
3. To understand and use the JavaScript concepts to design the interactive web pages.
4. To learn PHP to design server-side scripts for designing dynamic web pages.
5. To learn and understand the database connectivity with MYSQL, JDBC and JSP.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	To understand and apply the concepts of world wide web and HTML Tags for structuring web pages	Apply
CO2	To style the web pages using cascading style sheets (CSS) and various CSS rules and syntax	Apply
CO3	To implement client-side programming using JavaScript for creating interactive web pages	Apply
CO4	To implement server-side programming using PHP for designing dynamic web pages	Apply
CO5	To implement MYSQL, JDBC, JSP and Servlets for database connectivity and designing complete website	Apply

List of Experiments

S. No	Description	Hours
1	To design a simple webpage using HTML basic tags- Block Level and Inline Elements, Lists, Tables, Images.	2
2	To create a registration form in HTML using different input types and form elements.	2
3	To apply inline, internal and external CSS to the existing webpage for creating a stylish webpage.	2
4	To implement basic JavaScript programs that includes conditional statements, looping statements with the help of variables and operators.	2

5	To implement JavaScript Pop-Up Boxes, Functions, JavaScript Arrays, Array Methods, etc.	2
6	Implement event listeners (e.g., button clicks, keyboard and mouse events) and modify the DOM dynamically.	2
7	Design a responsive landing page with HTML, CSS, and basic interactivity, JavaScript and DOM.	4
8	To implement basic PHP programs, PHP functions using PHP variables and operators.	2
9	Implement database connectivity using PHP and MySQL.	4
10	To write and deploy basic Servlet program that includes servlet life cycle methods.	2
11	Install JDBC driver and connect Java application with the database.	2
12	To create JSP pages using scriptlets, expressions, and directives.	2
	Total Lab hours	**28**

Suggested Text Book(s):

1. Jon Duckett, HTML & CSS: Design and Build Websites (Wiley)
2. Jennifer Niederst Robbins, Learning Web Design (5th Edition), O'Reilly Media.
3. JSP: The Complete Reference by Phil Hanna

Suggested Reference Book(s):

1. Jon Duckett, JavaScript and JQuery: Interactive Front-End Web Development, Wiley.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (Cos) contribution to the Programme Outcomes (Pos)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Advanced Data Structures and Algorithms

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Data Structures and Algorithms

Course Objectives:

1. To understand core OOPs concepts like classes, objects, constructors, abstraction, and access specifiers.
2. To apply inheritance, polymorphism, and templates to real-world problems.
3. To implement arrays, stacks, queues, and linked lists for problem-solving.
4. To analyze and debug code with attention to outcomes and exceptions.
5. To design and implement class diagrams for practical applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain dynamic optimality: (Splay tree, Scape Goat Tree and Treap, Interval Tree, Segment Tree with Lazy Propagation Operations, Trees)	Understand
CO-2	Explain the importance of K-Dimensional Tree and R-trees	Apply
CO-3	Analyze the greedy method, divide-and-conquer, dynamic programming	Apply
CO-4	Understand the approximation algorithms for Steiner Tree, Vertex Cover, Set Cover, max-SAT, Subset Sum, Traveling Salesman Problem, etc.	Understand
CO-5	Analyze the Randomized Algorithms, Minimum Cut (Karger's Algorithm), Random Number Generator, Strong Password Suggester, Shuffle the Array, etc.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Review of Linear and Non-linear Data Structures: Stack, Queue, Tree, and Graph	**4**
2	Advanced Data Structures: Splay Tree, Scape Goat Tree and Treap, Disjoint Set, Interval Tree, Segment Tree with Lazy Propagation Operations, Priority Search Tree, Binary Indexed Tree, Range Tree and Range Queries, etc.	**10**
3	Introduction to Multi-Dimensional Data Structures: K-Dimensional Tree, Quad Tree, Octree, Binary Search Partitioning Trees, R-trees, Multidimensional B-trees ,etc.	**10**

4	Review of algorithm design techniques: greedy method, divide-and-conquer, dynamic programming with advanced applications, Algorithms for maximum flow	**4**
5	Approximation Algorithms for Steiner Tree, Vertex Cover, Set Cover, max-SAT, Subset Sum, Knapsack, Bin Packing, Scheduling, Traveling Salesman Problem, K-center, etc.	**10**
6	Randomized Algorithms for Quick Sort, Binary Search, Minimum Cut (Karger's Algorithm), Random Number Generator, Strong Password Suggester, Shuffle the Array, etc.	**4**
Total lectures		**42**

Suggested Text Book(s):

1. Brass, Peter. Advanced data structures. Vol. 193. Cambridge: Cambridge University Press, 2008.
2. La Rocca, Marcello. Advanced Algorithms and Data Structures. Simon and Schuster, 2021.
3. Saha, Suman, and Shailendra Shukla. Advanced data structures: theory and applications. Chapman and Hall/CRC, 2019.

Suggested Reference Book(s):

4. Cormen, Thomas H., et al. Introduction to algorithms. MIT press, 2022.

Other useful resource(s):

1. Link of MIT course contents:https://ocw.mit.edu/courses/6-851-advanced-data-structures-spring-2012/
2. Link to NPTEL course contents:https://www.youtube.com/playlist?list=PL54i8TI-dREZtNaEDi6pe5z8Vp6SRKS8G
3. Link to topics related to course: https://people.iith.ac.in/rogers/adsa/adsa.htm

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Advanced Data Structures and Algorithms Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-1

Pre-requisite: DSA withC/C++/Python Programming

Course Objectives:

1. Develop problem solving ability using Programming
2. Develop ability to design and analyze algorithms
3. Introduce students to data abstraction and advanced data structures
4. Develop ability to design and evaluate advanced data structures
5. Apply advanced data structure concepts to various examples and real life applications

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Understanding the Splay Tree, Scape, Priority Search Tree	Apply
CO2	Implement string data structures: k-Dimensional Tree, Quad Tree, Multidimensional B-trees	Apply
CO3	Implementation of problems solvable using design techniques of greedy method, divide-and-conquer, dynamic programming	Apply
CO4	Implement Approximation Algorithms for Vertex Cover, Set Cover, max-SAT, Subset Sum, Knapsack, Bin Packing, Scheduling, Traveling Salesman Problem	Apply
CO5	Implement Randomized Algorithms for Quick Sort, Binary Search, Minimum Cut (Karger's Algorithm), Random Number Generator, Strong Password	Apply

List of Experiments

S. No	Description	Hours
1	Implementation of Stack, Queue, Tree, Graph, etc.	**4**
2	Implementation of Advanced Data Structures: Splay Tree, Scape Goat Tree, Treap, Disjoint Set, Interval Tree, Segment Tree with Lazy Propagation Operations, Priority Search Tree, Binary Indexed Tree, Range Tree and Range Queries, etc.	**6**
3	Implementation of Multi-Dimensional Data Structures: K-Dimensional Tree, Quad Tree, Octree, Binary Search, Partitioning Trees, R-trees, Multidimensional B-trees, etc.	**6**
4	Implementation of problems solvable using design techniques of greedy method, divide-and-conquer, dynamic programming	**2**
5	Implementation of Approximation Algorithms for Steiner Tree, Vertex Cover, Set Cover, max-SAT, Subset Sum, Knapsack, Bin Packing, Scheduling, Traveling Salesman Problem, K-center, etc.	**6**

6	Implementation of Randomized Algorithms for Quick Sort, Binary Search, Minimum Cut (Karger's Algorithm), Random Number Generator, Strong Password Suggester, Shuffle the Array, etc.	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. Peter Brass: Advanced Data Structures
2. Cormen et al.: Introduction to Algorithms
3. Saha, Suman, and Shailendra Shukla. Advanced data structures: theory and applications. hapman and Hall/CRC, 2019.

Other useful resource(s):

4. Link of MIT course contents: https://ocw.mit.edu/courses/6-851-advanced-data-structures-spring-2012/
5. Link to NPTEL course contents: https://www.youtube.com/playlist?list=PL54i8TI-dREZtNaEDi6pe5z8Vp6SRKS8G
6. Link to topics related to course: https://people.iith.ac.in/rogers/adsa/adsa.htm

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Distributed and Cloud Computing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

1. Develop strong foundations in distributed system principles
2. Understand synchronization, election, agreement, and fault tolerance protocols
3. Explain consistency, replication, and recovery mechanisms
4. Correlate distributed systems with cloud computing paradigms
5. Introduce virtualization, cloud security, and AWS-based case studies

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain synchronization, logical clocks, event ordering, and global state collection.	Understand
CO-2	Analyze election, mutual exclusion, and deadlock algorithms.	Analyze
CO-3	Evaluate agreement, consistency, fault tolerance, and recovery.	Evaluate
CO-4	Correlate distributed principles with cloud computing and virtualization.	Analyze
CO-5	Apply AWS services (EC2, S3, RDS) for deployments	Apply

Course Contents:

Unit	Contents	Lectures required
1	**Foundations of Distributed Systems** Review of operating system principles, Theoretical foundations of distributed systems, Synchronization mechanisms, resource models Clock synchronization, logical and vector clocks, Event ordering and timestamp recording, Global state collection and snapshot algorithms	**6**
2	**Coordination, Election and Deadlocks** Election algorithms: Bully and Ring, Termination detection, Distributed mutual exclusion, Token and non-token based algorithms, Comparative performance analysis, Distributed deadlock detection and recovery	**8**
3	**Agreement, Consistency and Fault Tolerance** Agreement protocols and system models, Byzantine failures and solutions, Consistency and replication issues, Data-centric and client-centric consistency, Epidemic protocols, Distributed commit and failure recovery	**10**
4	**Cloud Computing and Virtualization** Cloud introduction and correlation with distributed systems, Deployment	**14**

	and service models, Essential characteristics and enabling technologies, Virtualization techniques, VMs, VMMs, Live migration and virtual clusters, Containers, Docker overview, Cloud data and network security, Access control and authentication	
5	**AWS Case Study** Elastic Compute Cloud (EC2), Simple Storage Service (S3), Relational Database Service (RDS), Use cases in distributed cloud applications	**4**
	Total lectures	**42**

Suggested Text Book(s):

1. Tanenbaum, A.S. and Van Steen, M. — Distributed Systems: Principles and Paradigms, 3rd Ed., Pearson, 2017
2. Coulouris, G., Dollimore, J., Kindberg, T., Blair, G. — Distributed Systems: Concepts and Design, 5th Ed., Addison-Wesley, 2011
3. Buyya, R., Vecchiola, C., Selvi, S.T. — Mastering Cloud Computing, McGraw-Hill, 2013

Suggested Reference Book(s):

5. Lynch, N.A. — Distributed Algorithms, Morgan Kaufmann, 1996
6. Birman, K. — Reliable Distributed Systems: Technologies, Web Services and Applications, Springer, 2005
7. Butrico, M., et al. — Cloud Computing: Principles and Paradigms (Buyya ed.), Wiley, 2011

Other useful resource(s):

1. AWS Documentation — https://docs.aws.amazon.com
2. MIT 6.824 Distributed Systems (Open Courseware) — https://pdos.csail.mit.edu/6.824
3. The Morning Paper (distributed systems papers) — https://blog.acolyer.org

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	–	–	–	–	1	–	**1.5**
CO-2	3	3	2	2	1	–	–	–	–	1	–	**2**
CO-3	3	3	2	3	1	–	–	1	–	1	–	**2**
CO-4	3	2	3	2	3	1	1	1	–	1	1	**1.8**
CO-5	3	2	3	1	3	–	–	–	1	1	1	**1.75**
Average	**3**	**2.4**	**2.2**	**1.8**	**1.8**	**1**	**1**	**1**	**1**	**1**	**1**	

Information Security and Cryptography

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Programming fundamentals, Operating system basics, Basic linux commands

Course Objectives:

1. To understand fundamental concepts of information security, threat landscape, and security models.
2. To apply modern cryptographic techniques including symmetric, asymmetric, and hashing mechanisms.
3. To analyze secure communication protocols and real-world security infrastructures such as PKI and TLS.
4. To evaluate system vulnerabilities, cyber attacks, and defense mechanisms in modern computing environments.
5. To design basic secure systems incorporating cryptographic tools and contemporary security practices.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain core concepts of information security, threats, vulnerabilities, and attack models.	Understand
CO-2	Apply cryptographic algorithms including symmetric, asymmetric, and hash-based techniques.	Apply
CO-3	Analyze security protocols such as SSL TLS, PKI, and secure communication mechanisms.	Apply
CO-4	Evaluate modern cyber threats including malware, web attacks, and network vulnerabilities.	Analyze
CO-5	Design and implement basic secure solutions using cryptographic and cyber security techniques.	Create

Course Contents:

Unit	Contents	Lectures required
1	Overview of Information Security – Security goals including confidentiality, integrity, availability, Threats, vulnerabilities, risk and attack surface, Sources of threats, Classification of attacks, Introduction to security frameworks and models	**4**
2	Fundamentals of Cryptography – Symmetric and asymmetric cryptography, Stream and block ciphers, Key management and key lifecycle, Diffie Hellman key exchange, Introduction to Public Key Infrastructure, Digital certificates	**4**
3	Hash Functions and Digital Signatures – Properties of cryptographic hash functions, SHA family including SHA 2 and SHA 3, Message authentication codes, HMAC, Digital signatures, Introduction to	**5**

	blockchain hashing concepts	
4	Practical Cryptography Tools – GPG basics, Key generation and management, Trust models, Command line usage, GUI tools such as Kleopatra and Seahorse, Introduction to OpenSSL and secure communication tools	**5**
5	Modern Block Ciphers and Encryption – Principles of block ciphers, Feistel networks, Substitution and permutation, AES in detail, Block cipher modes including CBC, CTR and GCM, Limitations of DES and 3DES, Authenticated encryption	**6**
6	Mathematical Foundations – Prime numbers, Modular arithmetic, Euclid algorithm, Fermat and Euler theorems, Discrete logarithm problem, Introduction to elliptic curve mathematics	**6**
7	Public Key Cryptography and Emerging Trends – RSA algorithm, Elliptic Curve Cryptography, Digital certificates and PKI architecture, Introduction to post quantum cryptography and zero knowledge proofs	**6**
8	Cyber Security Practices and Applications – Cyber crimes and classifications, Malware including ransomware and spyware, Web attacks such as SQL injection and cross site scripting, Network security basics, Cloud security fundamentals, Zero Trust Architecture, Multi factor authentication, Basics of ethical hacking and penetration testing	**6**
Total lectures		**42**

Suggested Text Book(s):

1. William Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education, 7th Edition, 2017.
2. Behrouz A. Forouzan and Debdeep Mukhopadhyay, Cryptography and Network Security, McGraw Hill Education, 3rd Edition, 2015.
3. Christof Paar and Jan Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 1st Edition, 2010.

Suggested Reference Book(s):

1. Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, Wiley India, 2nd Edition, 2015.
2. Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, CRC Press, 2nd Edition, 2014.

Other useful resource(s):

1. NPTEL, Cryptography and Network Security Courses, IITs and IISc, Available at https://nptel.ac.in
2. Coursera, Cryptography and Cyber Security Specializations, Various Universities, Available at https://www.coursera.org
3. edX, Cyber Security Professional Courses, Various Institutions, Available at https://www.edx.org
4. Cybrary, Cyber Security Training Platform, Available at https://www.cybrary.it
5. OWASP Foundation, Web Security Resources and Guidelines, Available at https://owasp.org
6. National Institute of Standards and Technology, Cryptographic Standards and Guidelines, Available at https://csrc.nist.gov

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Data Analytics Using R and Python

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Python Programming, Statistics, Probability

Course Objectives:

1. Gain a clear understanding of the basic concepts of data analytics and how data is used to solve real-world problems.
2. Learn to use R and Python for writing programs to clean, organize, and analyze data.
3. Develop skills to explore datasets and apply basic statistical techniques to draw meaningful conclusions.
4. Create clear and informative visualizations to represent data using popular R and Python libraries.
5. Work confidently with essential data analytics libraries such as NumPy, pandas, and tidyverse.
6. Apply analytical and predictive techniques to solve practical problems using real datasets.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of data analytics.	Understand
CO-2	Demonstrate the concepts of R & Python for data analytics.	Familiarity
CO-3	Apply advanced methods and their quantitative analysis for real-world problems.	Apply
CO-4	Apply statistical methods for hypothesis testing and inference problems.	Apply
CO-5	Analyze, visualize and interpret the results for useful insights.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Data Definitions and Analytical Programming Techniques: Introduction to Data Analytics, Elements, Variables, and Data categorization, Levels of Measurement, Introduction to analytical programming languages R & Python, and Installing Software & Setting up, Lists & Dictionaries, Functions & Packages, Data frame, Import and Export data, Data Preprocessing.	10
2	Parametric &Non Parametric Tests: Hypothesis Testing, Assumption Testing, T-Test, Power Analysis, ANOVA, Fitting ANOVA Model in Python & R, Wilcoxon Tests, Mann-Whitney U Test	6
3	Correlation & Association Analysis: Pearson Correlation, Spearman Correlation, Kendall Tau Correlation, Affinity Analysis & Market Basket	7

	Analysis APriori Algorithm, Association Rules, Frequent Pattern Analysis	
4	Data Analysis Techniques: Introduction to Machine Learning, Applications of ML Library in R & Python for Supervised & Unsupervised Learning, Basic Neural Network, Transfer Function Models, Multivariate Time Series Analysis.	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Navlani, A., Fandango, A. and Idris, I. (2021). Python Data Analysis – Third Ed. Packt Publishing Ltd.
2. Jake vanderPlas, Python Data Science Handbook – Essential Tools for Working with Data, O'Really
3. Media, 2022
4. David J. Pine, Introduction to Python for Science and Engineering, CRC Press, 2019.
5. Manoj Kumar Srivastava and Namita Srivastava, Statistical Inference – Testing of Hypotheses, PHI, 2014.
6. Kabacoff, Robert I. R in action: data analysis and graphics with R. Simon and Schuster, 2015.
7. Haider, M. (2015). Getting Started with Data Science: Making Sense of Data with Analytics. IBM Press.

Suggested Reference Book(s):

1. Doing Data Science, Straight Talk from The Frontline, Cathy O'Neil and Rachel Schutt, O'Reilly (2014).
2. Robert Johansson, Numerical Python – Scientific Computing and Data Science Applications with NumPy, SciPy and Matplotlib, Apress, 2019
3. Robert Sedgewick, Kevin Wayne, Robert Dondero, Introduction to Programming in Python: An Inter-disciplinary Approach, Pearson India Education Services Pvt. Ltd., 2016
4. Nelli, F., Python Data Analytics: with Pandas, NumPy and Matplotlib, Apress, 2018.
5. Wickham, H., &Grolemund, G. (2016). R for data science: import, tidy, transform, visualize, and model data. " O'Reilly Media, Inc.".

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106107220

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Data Analytics using R and Python Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming Language Fundamentals

Course Objectives:

1. An understanding of fundamental concepts of data analytics and its importance in real-world applications.
2. Proficiency with R and Python programming languages for data analysis.
3. Skills with data collection, cleaning, preprocessing, and transformation techniques.
4. Experience with exploratory data analysis and effective data visualization methods.
5. Knowledge with basic statistical methods and machine learning algorithms used in analytics.
6. Ability with interpreting analytical results for data-driven decision making.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO1	Apply fundamental concepts of data analytics to understand and analyze real-world datasets.	Apply
CO2	Use R and Python programming languages to perform data manipulation, cleaning, and preprocessing tasks.	Apply
CO3	Perform exploratory data analysis and data visualization to extract meaningful insights from data.	Apply
CO4	Implement basic statistical techniques and machine learning algorithms for data analysis and prediction.	Apply

List of Experiments

S. No	**Description**	**Hours**
1	Program to install, import, and use basic R and Python libraries (NumPy, Pandas, Matplotlib, ggplot2). Program to create and manipulate vectors, lists, matrices, and data frames in R.	**2**
2	Program to perform basic data types and control structures in Python. Program to read data from CSV/Excel files and display summary statistics.	**2**
3	Program to handle missing values (NA/NULL) using mean, median, or deletion methods.	**2**
4	Program to compute descriptive statistics (mean, median, mode, variance, SD). Program to perform data normalization and standardization.	**2**

5	Program to generate data visualizations: • Bar chart • Histogram • Box plot • Scatter plot	**2**
6	Program to detect and remove outliers using IQR/Z-score. Program to perform correlation and covariance analysis.	**2**
7	Program to filter, sort, and group data using Pandas and dplyr. Program to perform data aggregation and pivot operations.	**2**
8	Program to merge and join multiple datasets.	**2**
9	Program to apply string manipulation and date-time operations.	**2**
10	Program to perform feature selection and feature scaling.	**2**
11	Program to perform probability distribution analysis.	**2**
12	Program to implement hypothesis testing (t-test, z-test, chi-square test).	**2**
13	Project	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. Wes McKinney, Python for Data Analysis, O'Reilly Media.
2. Hadley Wickham & Garrett Grolemund, R for Data Science, O'Reilly Media.
3. Anil Maheshwari, Data Analytics Made Accessible, McGraw Hill.
4. Gareth James et al., An Introduction to Statistical Learning, Springer.
5. Link to topics related to course:

vi. https://www.bigbookofr.com/

vii. https://rstudio-education.github.io/hopr/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Mobile Application Development

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Basic Programming (C/C++/Java), OOP Concepts

Course Objectives:

1. To understand mobile application development concepts, platforms, and architecture.
2. To develop basic Android applications using activities, intents, and UI components.
3. To design user-friendly interfaces using layouts and widgets.
4. To implement event-driven programming in mobile applications.
5. To integrate database and device features in mobile applications.
6. Understand basic mobile application security practices, including permissions and secure data handling.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the mobile application development lifecycle, Android architecture, and core components	Understand
CO-2	Apply concepts of activities, intents, and event handling to develop simple apps	Apply
CO-3	Design user interfaces using layouts, widgets, and navigation components	Create
CO-4	Analyze and implement advanced UI components, data handling techniques, and basic application security practices	Analyze
CO-5	Develop mobile applications integrating databases, device services, and secure data handling mechanisms	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Mobile App Development: Mobile platforms overview (Android), app development lifecycle, Android architecture, setting up development environment (Android Studio, SDK), project structure, Activities, Activity lifecycle, Intents (explicit & implicit), manifest file, passing data between activities	**5**
2	Event Handling and Interaction: Event-driven programming, handling button clicks, listeners (onClick, onLongClick), input handling, simple app development using event listeners	**4**

3	User Interface Design – I: Views and ViewGroups, Layouts (Linear, Relative, Constraint), TextView, EditText, ImageView, ListView, GridView, Spinner, XML-based UI design, Adapter classes	**6**
4	User Interface Design – II: Advanced UI components: RadioButton, CheckBox, ToggleButton, ProgressBar, Menus, Navigation (Tabs, Navigation Drawer basics), Date & Time Picker, handling gestures (swipe, pinch, zoom)	**5**
5	Advanced UI and Device Features: Customizing ListView, GridView, RecyclerView (intro), filtering data, sending SMS/email (conceptual), introduction to location-based services and Google Maps API	**4**
6	Mobile Database (SQLite): Introduction to SQLite, database creation, CRUD operations, content values, simple app with database integration, data persistence. Application security fundamentals: Android permissions model, runtime permissions, secure storage of user data, data privacy principles, secure coding practices.	**4**
Total lectures		**28**

Suggested Text Book(s):

1. Wei-Meng Lee, Beginning Android Programming, Wiley
2. Bill Phillips et al., Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch

Suggested Reference Book(s):

1. Reto Meier, Professional Android Development, Wiley
2. Jerome (J.F.) DiMarzio, Android: A Programmer's Guide, McGraw-Hill

Other useful resource(s):

1. Android Developer Documentation: https://developer.android.com
2. NPTEL Courses on Mobile Computing / Android Development: https://nptel.ac.in/courses/106106147

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered up to T-1
2	T-2	25	1.5 Hours	Syllabus covered up to T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	2	1	2	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	1	2	**1.9**
CO-3	3	2	3	2	3	1	1	2	3	1	2	**2.0**
CO-4	2	3	2	3	3	1	2	1	2	2	2	**2.1**
CO-5	3	2	3	2	3	1	2	2	3	2	3	**2.4**
Average	**2.8**	**2.4**	**2.2**	**2**	**2.8**	**1**	**1.4**	**1.4**	**2.4**	**1.4**	**2.2**	

Mobile Application Development Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basic Programming, OOP Concepts

Course Objectives:

1. To provide hands-on experience in developing Android applications.
2. To implement activities, intents, and event-driven programming.
3. To design interactive user interfaces using layouts and widgets.
4. To develop applications using mobile databases (SQLite).
5. To integrate device features and services into mobile applications.
6. Implement basic mobile application security practices, including permission handling and secure data storage.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Develop basic Android applications using activities and intents	Apply
CO-2	Implement event handling and user interaction in mobile apps	Apply
CO-3	Design user interfaces using layouts, views, and adapters	Create
CO-4	Develop applications with advanced UI components, device features, and secure handling of permissions	Apply
CO-5	Implement database-driven applications using SQLite and basic secure data handling practices	Apply

List of Experiments

S. No	Description	Hours
1	Installation and setup of Android Studio, SDK, emulator; creating the first Android application (Hello World)	**2**
2	Study of Android project structure, Activity lifecycle demonstration using a sample app	**2**
3	Developing an application using Intents (explicit and implicit) and passing data between activities	**2**
4	Event handling: Button clicks, listeners (onClick), simple calculator application	**2**
5	Designing UI using Layouts (Linear, Relative, Constraint) with XML	**2**
6	Implementing UI elements: TextView, EditText, and ImageView with user input handling	**2**

7	Implementing ListView/GridView using Adapter classes	**2**
8	Creating an application using a Spinner and handling item selection events	**2**
9	Implementing advanced UI controls: RadioButton, CheckBox, ToggleButton, ProgressBar	**2**
10	Working with Menus and Navigation (Options menu / Context menu)	**2**
11	Designing an app with DatePicker and TimePicker	**2**
12	Customizing ListView/introduction to RecyclerView	**2**
13	SQLite database: Create database, insert and retrieve data (simple CRUD app). Implement basic secure storage practices (e.g., avoiding hardcoded data, using internal storage/SharedPreferences securely)	**2**
14	Mini Application: Develop a small Android app integrating UI + database (e.g., student record/notes app). Include runtime permission handling (e.g., storage/location) and basic secure data handling	**2**
	Total Lab hours	**28**

Suggested/Resources:

1. Wei-Meng Lee, Beginning Android Programming, Wiley
2. Bill Phillips et al., Android Programming: The Big Nerd Ranch Guide

Other useful resource(s):

1. Android Developer Documentation: https://developer.android.com

Evaluation Scheme:

S. No	**Exam**	**Marks**
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**Average**
CO-1	3	2	2	1	3	1	1	1	2	1	2	**1.7**
CO-2	3	2	2	2	3	1	1	1	2	1	2	**1.8**
CO-3	3	2	3	2	3	1	1	2	3	1	2	**2.1**
CO-4	3	2	3	2	3	1	2	2	2	1	2	**2.1**
CO-5	3	3	3	2	3	1	2	2	2	2	3	**2.4**
Average	**2.5**	**2.2**	**2.6**	**1.8**	**3**	**1**	**1.4**	**1.6**	**2.2**	**1.2**	**2.2**	

Fundamentals of Smart System and IoT

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basic Programming Knowledge, Basic Digital Electronics, Basic Computer Networking

Course Objectives:

10. Develop a foundational understanding of the Internet of Things (IoT), including smart sensors, communication paradigms, and their role in smart applications such as smart homes, smart cities, healthcare, and industrial automation.
11. Explain the architecture, communication protocols, and core components of IoT systems, along with the operational principles of Arduino microcontroller-based platforms.
12. Apply sensor interfacing techniques and Arduino programming skills to design, implement, and control basic IoT-enabled smart devices.
13. Analyze the performance characteristics, limitations, and suitability of different sensors and communication modules for various IoT-based smart system applications.
14. Demonstrate the design and working of real-world IoT applications in domains such as smart homes, healthcare, agriculture, and intelligent transportation systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Define the basic concepts of the Internet of Things (IoT), smart sensors, and their applications in various smart systems.	Understand
CO-2	Explain the architecture, protocols, and components of IoT systems and describe the working of Arduino microcontroller-based setups.	Understand
CO-3	Apply the knowledge of different sensors and Arduino programming to interface and control smart devices.	Apply
CO-4	Analyze the characteristics and performance of various sensors and communication modules used in IoT-based smart systems.	Apply
CO-5	Illustrate the design and operation of IoT applications in fields such as smart homes, healthcare, agriculture, and transportation.	Analyze

Course Contents:

Unit	Contents	Lectures required

1	Introduction to Smart Sensors and IoT: Introduction to IoT: Definition, evolution, enabling technologies. Smart Sensors: concept, role in IoT. Measuring and monitoring environmental conditions. Types of smart systems and application domains (home, healthcare, agriculture, transport)	**3**
2	IoT Architecture, Protocols, and Arduino Microcontroller IoT system architecture (device, gateway, cloud, application layers). Bread board and its internal connections. IoT protocol stack: Physical, MAC, Network, Transport, Application layers. Overview of common IoT protocols. Arduino overview: board components, microcontroller basics. Arduino development environment and basic operation.	**5**
3	Sensors and Their Characteristics: Sensor classification and working principles. Capacitive, resistive, SAW sensors. Sensors for temperature, pressure, humidity, gas. Static characteristics: sensitivity, accuracy, resolution. Dynamic characteristics: response time, bandwidth. Smart sensor architecture and interfacing challenges.	**5**
4	Design of Smart Systems: Embedded system design using Arduino. LED, tricolor LED, LCD, seven-segment display concepts. Push button interfacing. Pulse Width Modulation (PWM). Environmental sensors (DHT11).	**5**
5	Wireless Smart Systems: ESP8266-01 Wi-Fi module. IoT system components for cloud connectivity. ThingSpeak platform overview. MQTT protocol fundamentals. Introduction to low-level programming (AVR-GCC concepts).	**5**
6	IoT Applications: Smart Home systems: components, benefits, challenges. Smart Cities and Intelligent Transportation Systems. Smart Healthcare and wearable systems. Smart agriculture and precision farming. Smart factories and Industry 4.0. Societal, ethical, and privacy challenges	**5**
	Total lectures	**28**

Suggested Text Book(s):

1. Advances in Modern Sensors; Physics, design, simulation and applications by Sinha, G, R, IOP (Institute of Physics Publishing), 2020
2. Internet of Things: Architecture and Design Principles, Raj Kamal, McGrawHill. 2017
3. JanHo"ller,VlasiosTsiatsis,CatherineMulligan,Stamatis,Karnouskos,StefanAvesand,David Boyle, "From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence",Elsevier,2014.

Suggested Reference Book(s):

1. ArshdeepBahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things", CISCO Press, 2017.

Other useful resource(s):

1. https://spoken-tutorial.org/tutorial-search/?search_foss=Arduino&search_language=English
2. https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Fundamentals of Smart System and IoT Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basic Programming Knowledge, Basic Digital Electronics, Basic Computer Networking

Course Objectives:

1. Introduce students to the fundamental concepts of the Internet of Things (IoT) through hands-on interaction with embedded hardware, sensors, and actuators.
2. Enable students to acquire practical skills in interfacing sensors, displays, and actuators using microcontroller platforms such as Arduino and ESP32.
3. Familiarize students with basic networking concepts and cloud connectivity by implementing Wi-Fi–enabled data transmission and cloud-based data visualization.
4. Develop an understanding of edge computing concepts through Linux-based single-board computers by configuring Raspberry Pi for data acquisition and local storage.
5. Prepare students to design and analyze simple end-to-end IoT systems by integrating sensing, communication, data processing, and visualization components.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO1	To identify and explain basic components of IoT systems	Understand
CO2	To interface sensors, displays, and actuators using microcontroller platform	Apply
CO3	To implement wired and wireless IoT systems for sensor data transmission	Apply
CO4	To configure edge-based IoT systems using Raspberry Pi	Apply

List of Experiments

S. No	**Description**	**Hours**
1	Arduino IDE Setup and Digital I/O Test: Blink LED and Serial Monitor Output	**2**
2	Push Button and LED Control on Arduino: Toggle/Counter with Debouncing	**2**
3	Analog Input using Potentiometer/LDR: Sensor Value Reading and Mapping in Arduino	**2**
4	LCD Interfacing with Arduino: Display Sensor Readings (Potentiometer/LDR/DHT11)	**2**
5	PWM with Arduino: LED Brightness Control using Potentiometer (AnalogRead to AnalogWrite)	**2**

6	DHT11 Sensor Interfacing with Arduino: Temperature/Humidity Readings and Serial Logging	**2**
7	ESP8266 Wi-Fi Interfacing: Connect to Wi-Fi, Obtain IP, and Test Connectivity (Ping/Basic Web Request)	**2**
8	IoT Data Upload using ESP8266 and DHT11: Send Sensor Data to ThingSpeak using HTTP GET/POST	**2**
9	Cloud Dashboard on ThingSpeak: Visualize DHT11 Data, Field Configuration, and Update-Rate Analysis	**2**
10	Raspberry Pi Setup and GPIO with Python: Blink LED and Read Push Button (RPi.GPIO / gpiozero)	**2**
11	Raspberry Pi as IoT Gateway: Read DHT11/DS18B20 and Store Data in CSV/SQLite with Timestamp	**2**
12	Mini IoT Project (ESP8266/Raspberry Pi): Sensor Node and Cloud/Local Dashboard (Smart Home/Agriculture/Health)	**6**
	Total Lab hours	**28**

Suggested/Resources:

1. Advances in Modern Sensors; Physics, design, simulation and applications by Sinha, G, R, IOP (Institute of Physics Publishing), 2020
2. Internet of Things: Architecture and Design Principles, Raj Kamal, McGrawHill. 2017
3. Jan Ho"ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand, David Boyle, "From Machine-to-Machine to the Internet of Things -Introduction to a New Age of Intelligence", Elsevier, 2014.
4. ArshdeepBahga, Vijay Madisetti, "Internet of Things – A hands-on approach", Universities Press, 2015
5. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things", CISCO Press, 2017.
6. Link to topics related to course:

viii. https://spoken-tutorial.org/tutorial-search/?search_foss=Arduino&search_language=English
ix. https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet
x. https://www.arduino.cc/en/Tutorial/HomePage
xi. https://www.raspberrypi.org/documentation/
xii. "InternetofThings:AHands-onApproach",byArshdeepBahgaandVijayMadisetti

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Introduction To Compiler Design

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Programming in C/C++/Java, Data Structures

Course Objectives:

1. To understand the structure and working of compilers, including various phases involved in program translation.
2. To apply lexical and syntax analysis techniques using formal methods such as regular expressions and context-free grammars.
3. To analyze parsing strategies and error handling mechanisms used in compiler design.
4. To design intermediate representations and syntax-directed translation schemes for source programs.
5. To evaluate code generation and optimization techniques for improving execution efficiency of programs

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the phases of a compiler and their roles in program translation.	Understand
CO-2	Apply lexical analysis techniques using regular expressions and finite automata.	Apply
CO-3	Analyze different parsing techniques for syntax analysis.	Analyze
CO-4	Design intermediate code representations and syntax-directed translation schemes for source programs	Create
CO-5	Evaluate basic code generation and optimization techniques for improving program efficiency.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Compilers:** Compiler structure, phases of compiler, lexical, syntax, semantic analysis overview, compiler vs interpreter, tools and applications	**3**
2	**Lexical Analysis:** Role of lexical analyzer, tokens, patterns, lexemes, input buffering, regular expressions, finite automata, lexical analyzer generators (Lex/Flex)	**6**
3	**Syntax Analysis:** Context-free grammars, parse trees, ambiguity, top-down parsing (recursive descent, LL(1)), bottom-up parsing, shift-reduce conceptOperator Grammar, Operator Precedence Parsing, Augmented Grammar, LR(0) Item Set, LR(0) Parsing, LR(1) Item Set, SLR(1) Parsing, Canonical LR Parser, LALR parsing.	**8**
4	**Semantic Analysis & Intermediate Code:** Syntax-directed translation, type checking, Top-down, bottom-up evaluation, S-attributed vs L-attributed SDT, intermediate code generation, Three Address Code, Quadruple & Triples.	**5**

5	**Code Generation & Optimization:** Basic code generation concepts, target machine considerations, basic blocks, simple optimization techniques	**4**
6	**Run-time Environment & Symbol Table:** Memory allocation, activation records, parameter passing, symbol table organization	**2**
Total lectures		**28**

Suggested Text Book(s):

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, *Compilers: Principles, Techniques, and Tools*, Pearson
2. Kenneth C. Louden, *Compiler Construction*, Cengage Learning
3. Andrew W. Appel, *Modern Compiler Implementation in C*, Cambridge University Press

Suggested Reference Book(s):

1. John R. Levine, Tony Mason, Doug Brown, *Lex &Yacc*, O'Reilly
2. Steven S. Muchnick, *Advanced Compiler Design and Implementation*, Morgan Kaufmann

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106108113
   (Compiler Design – IIT Kharagpur)
2. Link to NPTEL course contents: https://nptel.ac.in/courses/106104123
   (Programming Languages and Compiler Design – IIT Madras)
3. Link to NPTEL course contents: https://nptel.ac.in/courses/106106131
   (Theory of Computation (supporting concepts))
4. **Evaluation Scheme:**

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	1	1	1	1	1	1	1.4
CO-2	3	3	1	2	3	1	1	1	1	1	1	1.7
CO-3	3	3	2	2	2	1	1	1	1	1	1	1.9
CO-4	3	2	3	2	2	1	1	1	2	1	1	2.0
CO-5	3	3	2	2	3	1	1	1	2	1	2	2.2
Average	**3.0**	**2.6**	**1.8**	**1.8**	**2.2**	**1.0**	**1.0**	**1.0**	**1.4**	**1.0**	**1.2**	

Introduction to Compiler Design Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basic knowledge of programming (C/C++)

Course Objectives:

1. To understand the fundamental phases of compiler design through practical implementation.
2. To apply concepts of lexical analysis, syntax analysis, and parsing techniques.
3. To design and implement various compiler components such as scanners and parsers.
4. To develop intermediate and target code generation techniques.
5. To gain hands-on experience with compiler construction tools and algorithms.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the different phases of a compiler and their functionalities.	Understand
CO-2	Implement lexical analysis techniques to identify tokens from source code.	Apply
CO-3	Analyze context-free grammars and construct parsing mechanisms for syntax analysis.	Analyze
CO-4	Develop intermediate code and implement parsing techniques for given languages.	Create
CO-5	Evaluate different parsing and code generation strategies for efficient compiler design.	Evaluate

List of Experiments

S. No	Description	Hours
1	Write a program to convert a given number into words and implement infix to postfix conversion.	2
2	Implement a Deterministic Finite Automaton (DFA) for a regular expression.	2
3	Develop a program to perform lexical analysis and identify tokens such as keywords, identifiers, operators, and constants.	2
4	Write a program to compute FIRST and FOLLOW sets of context-free grammar.	2
5	Write a program to detect and eliminate left recursion from a grammar.	2
6	Write a program to perform left factoring on a grammar.	2
7	Construct a predictive parsing table for a given grammar.	2
8	Implement a recursive descent parser for a grammar.	2
9	Implement shift reduce parsing technique.	2

10	Implement operator precedence parsing.	**2**
11	Implement LR parsing (SLR/CLR concept-based implementation).	**2**
12	Generate intermediate code (three address code) for expressions.	**2**
13	Implement basic code optimization techniques (constant folding, dead code elimination, etc.).	**2**
14	Implement simple code generation for a target machine (assembly-like output).	**2**
	Total Lab Hours	**28**

Suggested Books/Resources:

1. Compilers: Principles, techniques and tools, Aho, Sethi and Ullman
2. Compiler design in C, Holub
3. Advanced compiler design and implementation, Muchnick, Morgan and Kauffman

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106105190
   (Compiler Design – covers lexical analysis, parsing, syntax-directed translation, etc.)
2. Link to NPTEL course contents: https://nptel.ac.in/courses/106104123
   (Programming Languages and Translators – useful for compiler phases and implementation concepts)
3. Link to NPTEL course contents: https://nptel.ac.in/courses/106101224
   (Formal Languages and Automata Theory – helpful for DFA, CFG, parsing concepts)

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	0	0	0	1	0	1	**0.9**
CO-2	3	3	2	2	3	0	0	1	1	1	1	**1.6**
CO-3	2	3	2	3	2	0	0	1	1	1	2	**1.5**
CO-4	3	3	3	2	3	0	0	1	2	1	2	**1.8**
CO-5	2	2	2	2	2	1	1	1	2	2	3	**1.8**
Average	**2.6**	**2.6**	**2.0**	**2.0**	**2.2**	**0.2**	**0.2**	**0.8**	**1.4**	**1.0**	**1.8**	

Data Engineering

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Data Science

Course Objectives:

1. To understand foundational concepts of data engineering, including lifecycle, roles, and ecosystem components.
2. To explain and interpret data architectures and distributed systems used in modern data platforms.
3. To apply data engineering techniques to design and implement data pipelines for ingestion, transformation, and storage.
4. To analyze data workflows and system performance to identify bottlenecks and optimize processing efficiency in context with security principles.
5. To evaluate data engineering solutions based on scalability, reliability, security, and cost considerations.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand fundamentals of data engineering	Understand
CO-2	Apply architectural principles in system design	Apply
CO-3	Analyze data pipelines with security and workflows	Apply
CO-4	Evaluate data systems for efficiency and governance	Analyze
CO-5	Design scalable and reliable data engineering solutions	Create

Course Contents:

Unit	Contents	Lectures required
1	**Foundations of Data Engineering:** Definition, scope, and evolution of data engineering, Roles and responsibilities of data engineers, Data Engineering Lifecycle (conceptual framework), Data maturity models, Relationship with data science and software engineering, Introduction to DataOps principles	6
2	**Data Architecture and Systems:** Principles of data architecture design Batch vs streaming architectures, Data lakes, warehouses, and lakehouses, Distributed systems fundamentals, Storage and compute trade-offs, Technology selection criteria	8
3	**Data Engineering Lifecycle Implementation**: Data ingestion (batch and streaming), Data transformation (ETL/ELT processes), Data modeling and schema design, Workflow orchestration, Data serving and	8

	querying, Pipeline optimization. Role of security in data pipelines, Data lifecycle security, Security principles: CIA (Confidentiality, Integrity, Availability), Threat modeling for data systems, Data Anonymization	
4	**Governance, Optimization, and Advanced Concepts:** Data governance and quality management, Security, privacy, and compliance Observability and reliability, Cost optimization strategies (FinOps), Data mesh and modern data stack, Emerging trends in data engineering	6
Total lectures		**28**

Suggested Text Book(s):

1. Fundamentals of Data Engineering, Joe Reis and Matt Housley, O'Reilly Media, 2022.

Suggested Reference Book(s):

1. Designing Data-Intensive Applications, Martin Kleppmann, O'Reilly Media, 2017.
2. Data Engineering Fundamentals A Step by Step Approach, Sandeep Kumar Pandey, Notion Press, 2024.
3. Fundamentals of Analytics Engineering, Packt Publishing, 2024

Other useful resource(s):

1. Link to Coursera course contents: https://www.coursera.org/learn/introduction-to-data-engineering
2. Link to Youtube channel: https://www.youtube.com/watch?v=5LhU9hc0Rpw

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.0**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	3	3	2	2	3	1	1	1	2	2	2	**2.0**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3**	**2.4**	**2.4**	**2.0**	**3**	**1.0**	**1.0**	**1.0**	**2**	**2**	**2**	

Data Engineering lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Python

Course Objectives:

1. To develop a clear understanding of end-to-end data pipelines, including data ingestion, processing, storage, and visualization.
2. To design and implement relational and non-relational databases; perform CRUD operations and optimize queries.
3. To build Extract–Transform–Load pipelines using appropriate tools and frameworks for batch and real-time data processing.
4. To work with distributed computing frameworks to process large-scale datasets efficiently.
5. To implement validation checks and monitoring mechanisms to ensure data reliability and integrity.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO1	Design and implement end-to-end data pipelines for structured and unstructured data.	Apply
CO2	Apply distributed computing frameworks for processing large-scale datasets	Apply
CO3	Design data models suitable for analytical and reporting purposes.	Apply
CO4	Implement validation, monitoring, and error-handling mechanisms in data pipelines with security.	Apply
CO5	Apply acquired knowledge to solve real-world data engineering problems through hands-on projects	Apply

List of Experiments

S. No	**Description**	**Hours**
1	Fetch and store data from public APIs and convert into usable formats.	**2**
2	Create relational database schemas and populate data using MySQL/PostgreSQL	**2**
3	Store and retrieve data using NoSQL databases such as MongoDB.	**2**
4	Build an ETL pipeline to extract data, transform it, and load into a database	**2**
5	Perform basic data processing using Spark DataFrames.	**2**

6	Execute transformations and actions on large datasets using PySpark.	**2**
7	Design a star schema for a sample data warehouse.	**2**
8	Implement a simple data warehouse and load transformed data.	**2**
9	Simulate streaming data using tools like Kafka or Python streaming libraries.	**2**
10	Implement validation rules and error handling in pipelines.	**2**
11	Create dashboards using tools like Matplotlib/Power BI/Tableau.	**2**
12	WAP to Mask Sensitive Data (Data Anonymization)	**2**
13	Project	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. Designing Data-Intensive Applications – Martin Kleppmann
2. Data Engineering with Python – Paul Crickard
3. Streaming Systems – Tyler Akidau
4. Hadoop: The Definitive Guide – Tom White
5. Spark: The Definitive Guide – Bill Chambers and MateiZaharia

Other useful resource(s):

1. https://www.mongodb.com/docs/
2. https://hadoop.apache.org/docs/
3. https://spark.apache.org/docs/latest/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	3	1	1	1	1	1	2	**1.7**
CO-5	3	2	3	2	3	1	1	1	2	1	2	**1.9**
Average	**3.0**	**2.0**	**2.6**	**1.4**	**3**	**1.0**	**1.0**	**1.4**	**1.6**	**1.0**	**2.0**	

Foundation of Data Science and Visualization

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Linear Algebra, Python Programming

Course Objectives:

1. To introduce the principles and scope of Data Science
2. To develop skills in data preparation and pre-processing
3. To strengthen statistical and visualization competencies.
4. To familiarize learners with machine learning and network analysis

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Apply the Data Science Lifecycle	Understand
CO-2	Demonstrate Proficiency in Data Handling	Apply
CO-3	Utilize Statistical and Visualization Techniques	Apply
CO-4	Integrate Machine Learning and Network Analysis Concepts	Analyze

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Data Science** Scope and importance: Definitions, historical context, and engineering use cases. Data science lifecycle: Problem definition, data acquisition, preparation, analysis, modeling, deployment, monitoring. Types of data: Structured, semi-structured, unstructured; examples from sensors, logs, images, and text. Ethical and social implications: Privacy, bias, fairness, accountability, and responsible reporting.	**2**
2	**Data Preparation and Preprocessing forAnalytics** Data Sources and Structuring**:** Overview of data collection methods (surveys, sensors, logs, APIs, scraping, experiments).Data schemas and formats (relational, JSON/XML, columnar) with tidy data principles integrated.Introduction to ETL (extraction, transformation, loading; batch vs streaming). Data Quality Management**:**Missing values (types, diagnostics, imputation).Outliers (detection and treatment). Data Transformation and Pipelines**:**Scaling, normalization, encoding categorical variables, date/time parsing. Modular preprocessing pipelines and reproducibility (logging, versioning).	**5**

3	**Statistical Foundations** Descriptive statistics: Measures of central tendency, dispersion, shape, and robust summaries. Probability and distributions: Conceptual overview of discrete and continuous distributions and sampling. Inferential statistics: Confidence intervals, hypothesis testing, p-values, and practical interpretation. Correlation and regression: Pearson/Spearman correlation, simple linear regression, assumptions and diagnostics.	5
4	**Visualization Principles** Perception and cognition:Visual channels, pre-attentive features, Gestalt principles, and accessibility. Chart taxonomy: When to use bar, line, scatter, histogram, boxplot, violin, and heatmap**.** Design best practices:Axis design, labeling, annotation, color theory, and avoiding misleading visuals. Multi-dimensional visualization:Small multiples, faceting, dimensional encoding, and interactive considerations.	5
5	**Exploratory Data Analysis and Feature Engineering** EDA workflows: Question framing, iterative exploration, hypothesis generation, and documentation. Feature creation: Domain features, lag/rolling features for time series, interaction terms. Dimensionality reduction: Conceptual PCA, variance explained, and when to apply reduction. Bias variance tradeoff: Conceptual understanding and implications for model complexity.	4
6	**Introduction to Machine Learning and Model Evaluation** Supervised learning overview: Classification vs regression, common algorithms at a conceptual level. Train test methodology: Holdout, cross-validation, temporal splits for time series. Evaluation metrics: Accuracy, precision, recall, F1, ROC/AUC, RMSE, and metric selection by problem. Regularization and model selection: Overfitting, underfitting, L1/L2 intuition, and model comparison.	4
7	**Mining Social Networks and Graphs** Centrality measures: Degree, closeness, betweenness, eigenvector centrality, and interpretation. Community detection: Modularity, Louvain method, and practical considerations. Graph partitioning and neighborhoods: Cuts, ego networks, and local vs global structure.	3
	Total lectures	**28**

Suggested Text Book(s):

1. Foundations of Data Science – Santosh Kumar Sahu, HerisonSurbakti, Ismail Keshta, HaewonByeon Introductory text covering scope, lifecycle, and applications of data science.
2. Data Science from Scratch: First Principles with Python” – Joel Grus
3. Storytelling with Data: A Data Visualization Guide for Business Professionals – Cole NussbaumerKnaflic
4. Practical Statistics for Data Scientists” – Peter Bruce, Andrew Bruce, and Peter Gedeck Covers descriptive statistics, probability, regression, and hypothesis testing in a data science context.

Suggested Reference Book(s):

1. Python Data Science Handbook – Jake VanderPlas
2. The Elements of Statistical Learning – Trevor Hastie, Robert Tibshirani, Jerome Friedman
3. Mining of Massive Datasets – Jure Leskovec, Anand Rajaraman, Jeffrey Ullman
4. Data Visualization: Storytelling Using Data – Sharada Sringeswara, Purvi Tiwari, U. Dinesh Kumar

5. Foundations of Data Science Lecture Notes – David Sterratt, Kobi Gal, Hiroshi Shimodaira, Iain Murray, Steve Renals (University of Edinburgh)

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	3	3	2	2	1	2	2	2	2	2.27
CO-2	3	2	3	2	3	2	1	2	2	2	1	2.09
CO-3	2	2	2	3	3	2	1	3	2	3	1	2.18
CO-4	3	2	2	2	3	2	1	2	2	2	1	2
Average	2.75	2.25	2.5	2.5	2.75	2	1	2.25	2	2.25	1.25	

Foundation of Data Science and Visualization Lab

COURSE CODE:
COURSE CREDITS: 1
CORE/ELECTIVE: Elective
L-T-P: 0-0-2

Pre-requisite: Python Programming

Course Objectives:

1. To establish reproducible data science practices
2. To develop competence in data acquisition and pre-processing
3. To strengthen statistical and visualization competencies.
4. To familiarize learners with machine learning and network analysis

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Develop Reproducible Data Science Workflows	Apply
CO-2	Use Advanced Data Handling and Pre-processing Skills	Apply
CO-3	Use Statistical and Visualization Techniques	Apply
CO-4	Integrate Machine Learning and Network Analysis Concepts	Apply

List of Experiments

S. No	Description	Hours
1	**Environment and Reproducible Workflow** Create a reproducible Python environment and a GitHub repository that contains a well-structured Jupyter notebook demonstrating environment checks and a tiny data analysis pipeline. **Requirements** • Create requirements.txt or environment.yml. • Initialize a Git repo, make at least three meaningful commits, and push to GitHub. • Notebook must run top-to-bottom without errors. **Input and output** • **Input:** none (create a small synthetic CSV inside notebook). • **Output:** GitHub repo URL; notebook env_check.ipynb; requirements.txt.	2
2	Write a Python program that reads a multi-sensor CSV with irregular timestamps, parses timestamps, resamples all sensors to a uniform frequency, aligns sensors, and writes the aligned CSV. **Requirements** • Parse ISO timestamps and handle timezone if present. • Resample to a user-specified frequency (e.g., 100 Hz, 1s). • Fill missing values using interpolation and document choices. **Input and output** • **Input:** sensors_raw.csv with columns timestamp, sensor1, sensor2,	2

	.... • **Output:** sensors_aligned.csv and a notebook showing plots of before/after.	
3	**Signal Denoising and Filtering** Implement and compare three denoising methods on a noisy accelerometer signal: moving average, median filter, and Butterworth low-pass filter. Quantify SNR improvement. **Requirements** • Implement filters as reusable functions. • Compute SNR before and after filtering. • Plot time domain and frequency domain comparisons. **Input and output** • **Input:** vibration_noisy.csv with time, accel. • **Output:** Filtered signals, SNR table, plots.	2
4	**FFT and Spectral Analysis** Write a Python program to compute the FFT and power spectral density of a vibration signal, identify the top three spectral peaks, and explain their likely physical origin. **Requirements** • Compute PSD with appropriate windowing and overlap. • Identify peak frequencies and amplitudes. • Demonstrate effect of window length on frequency resolution. **Input and output** • **Input:** vibration.csv with time, signal. • **Output:** PSD plots, list of top three peaks, short interpretation.	2
5	**Outlier and Fault Detection** Implement statistical fault detection on process data using IQR, z-score, and a CUSUM detector. Compare detection times and false alarm rates on injected faults. **Requirements** • Implement three detectors and a fault injection routine. • Compute detection latency and false positive rate for each method. **Input and output** • **Input:** process_log.csv with time, variable. • **Output:** Detection comparison table, plots of alarms.	2
6	**Missing Data and Stream Alignment** Simulate packet loss in a multi-sensor stream, implement three imputation strategies (forward fill, linear interpolation, spline), and evaluate reconstruction RMSE against original data. **Requirements** • Simulate random packet loss with configurable loss rate. • Implement and compare imputation methods quantitatively. **Input and output** • **Input:** multi_sensor.csv original stream. • **Output:** RMSE table, plots showing original, missing, and imputed signals.	2
7	**Control System Response Analysis** Given control loop logs, compute step response metrics (rise time, settling time, overshoot), plot responses for multiple PID settings, and recommend tuning changes. **Requirements** • Implement metric computations programmatically. • Compare at least three PID parameter sets. **Input and output** • **Input:** control_logs.csv with time, setpoint, pv, control. • **Output:** Metric table, overlay plots, tuning recommendations.	2
8	**Regression for Thermal Systems** Build a regression model to predict room temperature given ambient conditions and HVAC settings. Engineer lag features and evaluate model	2

	residuals for physical plausibility. **Requirements** • Create lag and rolling features. • Train linear and regularized models; compare performance. • Interpret coefficients in engineering terms. **Input and output** • **Input:** thermal_data.csv with time, ambient, hvac_power, room_temp. • **Output:** Model files, performance metrics, coefficient interpretation.	
9	**Image Inspection Pipeline** **Problem statement** Implement an image preprocessing and defect detection pipeline using classical computer vision to detect surface defects on manufactured parts. **Requirements** • Implement grayscale conversion, denoising, adaptive thresholding, morphological operations, and contour filtering. • Output annotated images with bounding boxes. **Input and output** • **Input:** Folder parts_images/. • **Output:** Annotated images saved to output/, detection summary CSV.	**2**
10	**PCA for Multi Sensor Data** Apply PCA to multi-channel accelerometer data to identify dominant modes, plot explained variance, and reconstruct signals using first k components. **Requirements** • Standardize channels, compute PCA, plot scree plot, reconstruct and compute reconstruction error. **Input and output** • **Input:** multi_accel.csv with channels a1..aN. • **Output:** Scree plot, reconstruction error table, reconstructed signals.	**2**
11	***Predictive Maintenance Model*** Create a predictive maintenance pipeline that extracts time and frequency features from rolling windows, trains a classifier to predict imminent failure, and evaluates using F1 and ROC. **Requirements** • Feature extraction (RMS, spectral energy, kurtosis, etc.). • Use time-aware cross validation; report precision, recall, F1, ROC AUC. **Input and output** • **Input:** run_to_failure.csv with time series and failure timestamps. • **Output:** Trained model, evaluation metrics, feature importance.	**2**
12	***Telemetry and Network Health Analysis*** Analyze telemetry logs from a distributed system, compute node health metrics, cluster nodes by behavior, and visualize a dashboard of node health. **Requirements** • Aggregate metrics per node, compute rolling anomalies, cluster nodes, and produce heatmap and network graph. **Input and output** • **Input:** telemetry_logs.csv with time, node, latency, throughput, errors. • **Output:** Cluster assignments, heatmaps, anomaly alerts CSV.	**2**
13	***Change Point Detection for Structures*** Implement and compare change point detection methods (CUSUM, offline Bayesian, and wavelet-based) on structural monitoring data with staged damage events. **Requirements** • Implement at least two methods; compute detection delay and false alarm rate; visualize detections.	**2**

	Input and output • **Input:** structure_monitor.csv with time, sensor. • **Output:** Detection comparison table, annotated time series plots.	
14	***Edge Model Prototype and Profiling*** Train a compact model for a classification task, export it in a lightweight format, and measure simulated inference latency and memory usage on constrained hardware profile. **Requirements** • Train compact model (e.g., small decision tree or tiny neural net). • Export model (joblib or ONNX), simulate inference loop, measure latency and memory, propose quantization/pruning steps. **Input and output** • **Input:** edge_dataset.csv. • **Output:** Exported model file, latency/memory report, suggested optimizations.	**2**
	Total Lab hours	**28**

Suggested/Resources:

1. Python for Data Analysis — Wes McKinney; pandas author; practical for labs and EDA.
2. Python Data Science Handbook — Jake VanderPlas; notebooks and examples for pandas, NumPy, scikit learn.
3. Practical Statistics for Data Scientists — Peter Bruce & Andrew Bruce; applied inference and CI.
4. Hands On Machine Learning with Scikit Learn, Keras, TensorFlow — AurélienGéron; end to end ML pipelines.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	2	3	2	1	1	2	2	2	1	**2**
CO-2	3	2	3	2	2	1	1	2	2	1	1	**1.81**
CO-3	3	2	2	3	3	2	2	3	2	2	1	**2.27**
CO-4	3	3	2	2	3	2	1	2	2	2	1	**2.09**
Average	**3**	**2.5**	**2.25**	**2.5**	**2.5**	**1.5**	**1.25**	**2.25**	**2**	**1.75**	**1**	**2.04**

Artificial and Computational Intelligence

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective-1

L-T-P: 2-0-0

Pre-requisite: Python Programming, Data Structures and Linear Algebra

Course Objectives:

1. Understand fundamental concepts of Artificial Intelligence and Computational Intelligence.
2. Apply optimization techniques to solve real-world problems.
3. Analyze different machine learning and fuzzy systems.
4. Evaluate intelligent systems and their performance.
5. Design intelligent models using computational intelligence techniques.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts, techniques, and applications of Artificial Intelligence and Computational Intelligence.	Understand
CO-2	Apply fuzzy logic concepts to solve decision-making problems.	Apply
CO-3	Analyze and implement artificial neural network models using appropriate learning algorithms.	Analyze
CO-4	Evaluate the performance of different AI and computational intelligence techniques using appropriate metrics.	Evaluate
CO-5	Design intelligent systems using computational intelligence approaches such as fuzzy logic, neural networks, or evolutionary algorithms.	Create

Course Contents:

Unit	Contents	Lectures required
1	AI definition, history and applications; AI vs Computational Intelligence; intelligent agents and their types; PEAS description; environment types (fully observable vs. partially observable, deterministic vs. stochastic, static vs. dynamic etc.); agent architectures: simple reflex agents, model-based reflex agents, goal-based agents.	**6**
2	Crisp vs Fuzzy sets, operations on fuzzy sets; membership functions; fuzzification; defuzzification; fuzzy inference systems (Mamdani, Sugeno), applications (e.g., temperature control).	**6**
3	Biological vs artificial neuron; perceptron; multilayer perceptron (MLP); activation functions; backpropagation algorithm; applications of neural network.	**5**

4	Evolutionary Algorithms: introduction to evolutionary computing; genetic algorithms; selection, crossover, mutation; fitness function design; convergence and performance issues; applications of genetic algorithms.	**6**
5	Swarm Intelligence: concept of swarm intelligence; particle swarm optimization (PSO); ant colony optimization (ACO); comparison with evolutionary algorithms; applications in optimization problems.	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Fourth Edition, Pearson Education, 2020.
2. Andries P. Engelbrecht, "Computational Intelligence: An Introduction", Second Edition, John Wiley & Sons, 2007.
3. Simon Haykin, "Neural Networks and Learning Machines", Third Edition, Prentice Hall, 2009.
4. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", Third Edition, John Wiley & Sons, 2010.
5. Melanie Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 1998.

Suggested Reference Book(s):

1. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.
2. A.E. Eiben and J.E. Smith, "Introduction to Evolutionary Computing", 2nd Edition, Springer, 2015.
3. Patterson, "Introduction to Artificial Intelligence & Expert Systems", Pearson, 1st ed. 2015.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106105078
2. Link to NPTEL course contents: https://nptel.ac.in/courses/106106139
3. Link to topics related to course: https://nptel.ac.in/courses/106102220

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	1	1	1	1	1	2	**1.4**
CO-2	3	3	2	1	2	1	1	1	1	1	2	**1.6**
CO-3	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-4	2	3	2	3	3	1	1	1	1	1	2	**1.8**
CO-5	3	3	3	2	3	2	1	2	2	2	3	**2.4**
Average	**2.8**	**2.8**	**2.0**	**1.8**	**2.4**	**1.2**	**1.0**	**1.2**	**1.2**	**1.2**	**2.2**	

Artificial and Computational Intelligence Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: Elective

L-T-P: 0-0-2

Pre-requisite: Python Programming, Data Structures and Linear Algebra

Course Objectives:

1. To provide hands-on experience with AI and computational intelligence techniques.
2. To design and simulate fuzzy logic controllers.
3. To develop neural network models for classification/regression tasks.
4. To apply evolutionary and swarm intelligence techniques for optimization problems.
5. To develop practical skills using modern AI tools and libraries.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Develop fuzzy logic systems for real-world applications.	Apply
CO2	Develop neural network models using perceptron and multi-layer perceptron with backpropagation.	Analyze
CO3	Apply genetic algorithms for solving optimization problems and analyze convergence behavior.	Analyze
CO4	Implement swarm intelligence techniques such as PSO and ACO for optimization problems.	Create
CO5	Compare performance of AI, fuzzy, neural, and evolutionary techniques on real-world problems.	Evaluate

List of Experiments

S. No	Description	Hours
1	To design and simulate a Simple Reflex Agent for a vacuum cleaner environment using Python.	**2**
2	To design and implement a Model-Based Agent that maintains an internal state and performs decision-making in a vacuum cleaner environment using Python.	**2**
3	Design and visualization of fuzzy sets and membership functions (triangular, trapezoidal, Gaussian).	**2**
4	Design a fuzzy logic controller in Python using libraries like scikit-fuzzy or skfuzzy. Simulate and visualize the control behavior of the fuzzy logic system for a simple control problem (e.g., temperature control, speed regulation).	**4**

5	Implementation of Single Layer Perceptron for binary classification.	**2**
6	Implementation of Multilayer Perceptron (MLP) using Python libraries. Build and train a feedforward neural network for the classification task and visualize the training process and results.	**2**
7	To implement the Backpropagation algorithm for training a neural network on a classification dataset and perform error analysis.	**2**
8	To implement the Backpropagation algorithm for training a neural network on a regression dataset and analyze prediction errors.	**2**
9	Implementation of Genetic Algorithm (selection, crossover, mutation). Solving an optimization problem such as the knapsack problem or the traveling salesman problem using Genetic Algorithm and analyzing convergence of the algorithm.	**4**
10	Implementation of Particle Swarm Optimization (PSO) for function optimization.	**2**
11	Implementation of Ant Colony Optimization (ACO) for shortest path problem.	**2**
12	Implementation of Ant Colony Optimization (ACO) for solving Travelling Salesman Problem.	**2**
Total Lab hours		**28**

Suggested/Resources:

1. Simon Haykin, "Neural Networks and Learning Machines", 3rd Edition, Pearson.
2. Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", Fourth Edition, Pearson Education, 2020.
3. Patterson, "Introduction to Artificial Intelligence & Expert Systems", Pearson, 1st ed. 2015.
4. S. Rajasekaran and G. A. Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications", PHI.
5. Melanie Mitchell, "An Introduction to Genetic Algorithms", MIT Press.
6. Timothy J. Ross, "Fuzzy Logic with Engineering Applications", Wiley.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106105078
2. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc25_ge15/preview
3. Kaggle Datasets: https://www.kaggle.com/datasets
4. UCI Machine Learning Repository: https://archive.ics.uci.edu/ml

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	1	1	2	**1.5**
CO-2	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-3	3	3	3	2	3	1	1	1	1	1	2	**1.9**
CO-4	2	3	2	3	3	1	1	1	1	1	2	**1.8**
CO-5	3	3	3	2	3	2	1	2	2	2	3	**2.4**
Average	**2.8**	**2.8**	**2.4**	**2.0**	**2.8**	**1.2**	**1.0**	**1.2**	**1.2**	**1.2**	**2.2**	

Fundamentals of Soft Computing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Basic Mathematics, Programming Fundamentals

Course Objectives:

1. To understand the fundamental concepts of soft computing and its components.
2. To study fuzzy logic systems and their applications in decision-making.
3. To learn artificial neural networks and their learning mechanisms.
4. To explore evolutionary algorithms such as genetic algorithms for optimization.
5. To apply hybrid soft computing techniques to real-world problems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the concepts of soft computing, fuzzy logic, and neural networks	Understand
CO-2	Apply fuzzy logic principles to solve decision-making problems	Apply
CO-3	Implement artificial neural network models for classification and prediction	Apply
CO-4	Analyze optimization problems using genetic algorithms and evolutionary techniques	Analyze
CO-5	Design hybrid soft computing models for real-world applications	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Soft Computing:** Concept of Soft Computing, Difference between Hard and Soft Computing, Components of Soft Computing, Applications in engineering and AI	2
2	**Fuzzy Logic:** Fuzzy Sets, Membership Functions, Operations on Fuzzy Sets, Fuzzy Relations, Fuzzy Rules and Inference Systems, Defuzzification Methods	5
3	**Artificial Neural Networks (ANN):** Biological Neuron Model, ANN Architecture, Types of Neural Networks, Learning Rules, Perceptron, Multilayer Perceptron, Backpropagation Algorithm	6
4	**Genetic Algorithms (GA):** Introduction to Evolutionary Computation, Genetic Algorithm Components, Selection, Crossover, Mutation, Fitness Function, Applications of GA	4

5	**Hybrid Systems:** Neuro-Fuzzy Systems, Genetic-Fuzzy Systems, Applications in pattern recognition, optimization, and control systems	**4**
6	**Swarm Intelligence:** Introduction to Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Applications in optimization problems	**4**
7	**Applications of Soft Computing:** Case studies in image processing, medical diagnosis, robotics, and data mining	**3**
	Total lectures	**28**

Suggested Text Book(s):

1. S. N. Sivanandam& S. N. Deepa, Principles of Soft Computing, Wiley India
2. K. T. Atanassov, Intuitionistic Fuzzy Sets, Springer
3. J. S. R. Jang, C. T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice Hall
4. Simon Haykin, Neural Networks and Learning Machines, Pearson

Suggested Reference Book(s):

1. David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Pearson
2. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley
3. Laurene Fausett, Fundamentals of Neural Networks, Pearson

Other useful resource(s):

1. NPTEL: Soft Computing – https://nptel.ac.in
2. Coursera: Neural Networks and Deep Learning
3. TutorialsPoint: Soft Computing Tutorials

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Fundamentals of Soft Computing Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: Fundamentals of Soft Computing (Theory), Basic Programming (Python/MATLAB preferred)

Course Objectives:

1. To implement fuzzy logic systems for decision-making problems.
2. To design and simulate artificial neural networks.
3. To apply genetic algorithms for optimization problems.
4. To explore swarm intelligence techniques for real-world applications.
5. To develop hybrid soft computing models using modern tools.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Implement fuzzy logic systems including fuzzification and defuzzification	Apply
CO2	Design and train artificial neural networks using suitable tools	Apply
CO3	Apply genetic algorithms to solve optimization problems	Apply
CO4	Implement swarm intelligence techniques such as PSO and ACO	Apply
CO5	Develop hybrid soft computing models for real-world applications	Apply

List of Experiments

S. No	Description	Hours
1	Introduction to Python/MATLAB environment for Soft Computing (NumPy, SciPy, MATLAB toolbox basics)	2
2	Implementation of basic fuzzy set operations (union, intersection, complement) and membership functions	3
3	Design of fuzzy inference system (Mamdani model) for decision-making problems (e.g., temperature control)	3
4	Implementation of defuzzification techniques (centroid, bisector, mean of maxima)	2
5	Implementation of single-layer perceptron model	3

6	Implementation of Multilayer Perceptron (MLP) using backpropagation algorithm	3
7	Training ANN for classification problems (e.g., Iris dataset / plant leaf dataset)	3
8	Implementation of Genetic Algorithm for function optimization (selection, crossover, mutation)	3
9	Implementation of Particle Swarm Optimization (PSO) for optimization problems	3
10	Mini Project: Hybrid model (Neuro-Fuzzy / GA-ANN) for real-world problem (e.g., plant classification, prediction, optimization)	3
Total Lab hours		**28**

Suggested/Resources:

1. S. N. Sivanandam& S. N. Deepa, Principles of Soft Computing, Wiley
2. J. S. R. Jang, Neuro-Fuzzy and Soft Computing, Prentice Hall
3. Simon Haykin, Neural Networks, Pearson
4. Timothy J. Ross, Fuzzy Logic with Engineering Applications, Wiley
5. Python Libraries: NumPy, TensorFlow/Keras, Scikit-learn
6. MATLAB Fuzzy Logic Toolbox & Neural Network Toolbox

Other useful resource(s):

1. NPTEL Soft Computing Course – https://nptel.ac.in
2. Coursera – Neural Networks and Deep Learning
3. Scikit-learn Documentation – https://scikit-learn.org

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Fundamentals of Computer and Cyber Security

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Computer Fundamentals

Course Objectives:

1. To introduce fundamental concepts of computer and cyber security, including threats, vulnerabilities, and the CIA triad.
2. To understand and demonstrate cryptographic techniques for securing data and communication.
3. To apply security mechanisms to networks and operating systems, and analyze common attack scenarios.
4. To evaluate security policies, ethical issues, and governance frameworks in organizational and legal contexts.
5. To design basic security solutions and incident response strategies for emerging technologies such as cloud, IoT, and mobile systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Recall and explain key cybersecurity concepts, terminology, and models (CIA triad, threats, vulnerabilities).	Understand
CO-2	Demonstrate the use of cryptographic algorithms (AES, RSA, hashing, digital signatures) in securing communication.	Apply
CO-3	Analyze network/system vulnerabilities and propose defense mechanisms (firewalls, IDS/IPS, secure coding).	Analyze
CO-4	Evaluate cybersecurity policies, ethical dilemmas, and compliance requirements (IT Act, GDPR, ISO frameworks).	Evaluate
CO-5	Design and implement basic security solutions for emerging domains (cloud, IoT, mobile) and incident response plans.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Definition and scope of cyber security, CIA triad: Confidentiality, Integrity, Availability, Threats, vulnerabilities, exploits, Risk management and security models, Case studies of real-world breaches	**4**
2	**Cryptography Essentials:** Classical ciphers (Caesar, Vigenère), Symmetric encryption (DES, AES), Asymmetric encryption (RSA, ECC), Hash functions (SHA family, MD5), Digital signatures and certificates, Public Key Infrastructure (PKI), Applications in secure communication	**6**

3	**Network Security Fundamentals:** OSI and TCP/IP security layers, Firewalls: packet filtering, stateful inspection, Intrusion Detection/Prevention Systems (IDS/IPS), Virtual Private Networks (VPNs), Secure protocols: TLS, SSH, HTTPS, Common attacks: DoS, DDoS, MITM, sniffing, Network monitoring and traffic analysis	**6**
4	**System Security:** Operating system hardening (Windows/Linux), User authentication and access control, Secure coding practices, Application vulnerabilities: buffer overflow, SQL injection, XSS, Patch management and updates, Malware types and defense strategies	**4**
5	**Security Policies and Management:** Security policies and frameworks (ISO 27001, NIST), Cyber laws: Indian IT Act, GDPR basics, Ethical hacking and penetration testing, Governance, compliance, and audit processes, Incident reporting and accountability	**4**
6	**Emerging Issues in Cyber Security:** Cloud security fundamentals (shared responsibility model), IoT security challenges (device vulnerabilities, botnets), Mobile security (app permissions, sandboxing), AI and machine learning in threat detection, Incident response planning and disaster recovery, Future trends: quantum cryptography, zero-trust architecture	**4**
Total lectures		**28**

Suggested Text Book(s):

1. James Graham, Richard Howard, Ryan Olson – Cyber Security Essentials, CRC Press, Taylor & Francis Group, 2011.
2. William Stallings & Lawrie Brown – Computer Security: Principles and Practice, Pearson (5^{th} Edition), 2024.

Suggested Reference Book(s):

1. Charles P. Pfleeger& Shari Lawrence Pfleeger, Security in Computing, Pearson, 2015, 5th Edition.
2. David Kim & Michael G. Solomon, Fundamentals of Information Systems Security, Jones & Bartlett Learning, 2018, 3rd Edition
3. Mark Stamp, Information Security: Principles and Practice, Wiley, 2019, 3rd Edition

Other useful resource(s):

1. NIST Cyber security Framework (2024 update): https://www.nist.gov/cyberframework
2. OWASP (Open Web Application Security Project): https://owasp.org
3. SANS Institute Reading Room (whitepapers & case studies): https://www.sans.org/white-papers
4. CERT-In (Indian Computer Emergency Response Team): https://www.cert-in.org.in
5. MIT OpenCourseWare – Computer Systems Security: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-858-computer-systems-security-fall-2014/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	1	–	–	–	2	2	**1.5**
CO-2	3	2	2	2	3	–	–	–	–	2	2	**2.3**
CO-3	3	3	3	3	3	2	2	–	2	2	2	**2.5**
CO-4	2	2	1	1	1	3	2	3	2	3	2	**2.0**
CO-5	3	3	3	3	3	3	3	3	3	3	3	**3.0**
Average	**2.8**	**2.4**	**2.0**	**2.0**	**2.2**	**2.3**	**2.3**	**3.0**	**2.3**	**2.4**	**2.3**	

Fundamentals of Computer and Cyber Security Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Computer Fundamentals

Course Objectives:

1. To familiarize with cyber security tools, environments, and basic commands for system and network monitoring.
2. To implement cryptographic algorithms and security mechanisms through hands-on coding and demonstrations.
3. To examine network traffic, system vulnerabilities, and application flaws using practical experiments.
4. To assess the effectiveness of firewalls, secure protocols, and hardening techniques in controlled lab scenarios.
5. To develop and document incident response strategies, security policies, and prototype solutions for emerging technologies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Identify and recall cyber security tools, commands, and environments used in lab practice.	Understand
CO2	Apply cryptographic techniques (AES, RSA, hashing) to secure files and communications.	Apply
CO3	Analyze captured network traffic and system/application vulnerabilities using Wireshark, firewalls, and test web apps.	Analyze
CO4	Evaluate the effectiveness of OS hardening, secure protocols, and penetration testing methods.	Evaluate
CO5	Create incident response plans, draft security policies, and design prototype solutions for cloud, IoT, and mobile security.	Create

List of Experiments

S. No	Description	Hours
1	**System Reconnaissance & Vulnerability Scanning** • Install Kali Linux / security tools, • Perform basic system reconnaissance using Nmap, • Identify open ports and potential vulnerabilities.	**2**
2	**Classical Cryptography Implementation** • Implement Caesar cipher and Vigenère cipher in Python. • Encrypt and decrypt sample text messages.	**2**
3	**Symmetric Encryption (AES)** • Use Python's cryptography library to implement AES. • Demonstrate secure file encryption and decryption. • Compare performance with different key sizes.	**4**

4	**Asymmetric Encryption (RSA) & Digital Signatures** • Generate RSA key pairs. • Encrypt and decrypt messages using public/private keys. • Demonstrate digital signatures and certificate validation.	4
5	**Hashing & Integrity Verification** • Implement SHA-256 hashing. • Verify file integrity using hash comparison. • Explore password hashing with bcrypt.	2
6	**Packet Sniffing & Traffic Analysis** • Capture and analyze network traffic using Wireshark. • Identify HTTP, DNS, and TCP packets. • Detect anomalies in traffic patterns.	4
7	**Firewall Configuration & Secure Protocols** • Configure firewall rules using iptables or Windows Firewall. • Demonstrate secure login using SSH. • Enable HTTPS/TLS for a sample web server.	2
8	**Web Application Vulnerabilities (SQL Injection & XSS)** • Demonstrate SQL injection on a test web app. • Show XSS attack and mitigation techniques.	2
9	**Operating System Hardening (2 hours)** • Configure user accounts, permissions, and password policies. • Apply patches and updates. • Demonstrate malware detection basics.	2
10	**Incident Response & Emerging Trends** • Configure multi-factor authentication. • Simulate a breach scenario and draft an incident response plan. • Explore IoT device vulnerability scanning.	4
	Total Lab hours	**28**

Suggested/Resources:

1. James Graham, Richard Howard, Ryan Olson – Cyber Security Essentials, CRC Press, Taylor & Francis Group, 2011.
2. William Stallings & Lawrie Brown – Computer Security: Principles and Practice, Pearson (5th Edition), 2024.
3. Charles P. Pfleeger& Shari Lawrence Pfleeger, Security in Computing, Pearson, 2015, 5th Edition.
4. David Kim & Michael G. Solomon, Fundamentals of Information Systems Security, Jones & Bartlett Learning, 2018, 3rd Edition
5. Mark Stamp, Information Security: Principles and Practice, Wiley, 2019, 3rd Edition

Other useful resource(s):

1. Kali Linux Documentation – Official guide for penetration testing tools: https://www.kali.org/docs
2. Wireshark Labs – Packet analysis tutorials: https://www.wireshark.org/docs
3. OWASP Web Security Testing Guide – Hands-on web app vulnerability testing: https://owasp.org/www-project-web-security-testing-guide
4. Metasploit Unleashed (Offensive Security) – Free penetration testing labs: https://www.offensive-security.com/metasploit-unleashed

Evaluation Scheme:

S. No	Exam	Marks

1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	–	–	–	–	2	2	**1.8**
CO-2	3	2	2	2	3	–	–	–	–	2	2	**2.3**
CO-3	3	3	3	3	3	2	2	–	2	2	2	**2.5**
CO-4	2	2	2	2	2	3	2	3	2	3	2	**2.3**
CO-5	3	3	3	3	3	3	3	3	3	3	3	**3.0**
Average	**2.8**	**2.4**	**2.2**	**2.2**	**2.6**	**2.3**	**2.3**	**3.0**	**2.3**	**2.4**	**2.4**	

Data Mining and Data Warehousing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Data Structures, Compilers, Operating Systems, Computer Networks, Machine Learning and Genetic Algorithms.

Course Objectives:

1. To describe the concept of Data warehouse & its attributes
2. To study different data warehouse models, architectures and implementation
3. To understand the basic concept of data mining and its functionality
4. To understand the concept of classification techniques and its implementation
5. To understand the concept of association rules, different techniques and implementation details
6. To understand the concept of cluster analysis, anomaly detection and its usage and implementation details

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand data mining concepts, KDD process, data types, and preprocessing techniques.	Understand
CO-2	Understand data warehouse architecture, multi-dimensional models, schemas, and OLAP systems.	Understand
CO-3	Apply and evaluate classification techniques for solving predictive data mining problems.	Apply and Evaluate
CO-4	Analyze and implement association rule mining techniques to discover meaningful patterns in data.	Analyze
CO-5	Apply and assess clustering techniques for pattern discovery and data analysis.	Apply

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Data Mining, Motivation, Challenges, Origins of Data Mining, Applications, Issues, Data Mining Tasks, stages of Knowledge discovery in database, Data: Types of Data, Data Quality, Measures of Similarity and Dissimilarity, Data Pre-processing, Measures of Similarity and Dissimilarity, Exploring Different Datasets, Summary Statistics, Data Visualization.	**8**
2	**Architecture:** Basic Concept of Data Warehouse, DBMS vs. Data Warehouse, Data marts, Metadata, Multidimensional data model, Data Cubes, Schemas for Multidimensional Database: Stars, Snowflakes and	**5**

	Fact Constellations, Data Warehouse Architecture, ETL Process (Extraction, Transformation and Loading), Distributed and Virtual Data Warehouse, Data Warehouse Manager, OLTP, OLAP, MOLAP, HOLAP, types of OLAP servers.	
3	**Classification:** Basic Concepts and Preliminaries, Approach to Solving a Classification Problem, Decision Tree Induction: Iterative Dichotomiser 3, C4.5 Decision Tree Algorithms, Classification and Regression Trees, Model Overfitting, Evaluating Performance of binary and multi-class classifiers. Rule-Based Classifier, Ensemble Learning Methods: Bagging, Boosting, Staking and Voting-based classifiers, Class Imbalance Problem,	**5**
4	**Association Analysis:** Basic Concepts and Problem Definition, Frequent Itemset Generation, Rule Generation, Representation of Frequent Itemsets, FP-Growth Algorithm, Evaluation of Association Patterns, Sequential Patterns, Subgraph Patterns.	**5**
5	**Cluster Analysis:** Basic Concepts, Characteristics of Data, Clusters, Cluster Evaluation, Partitional Clustering: K-Means, K-Medoids, Hierarchical Clustering: Agglomerative and Divisive Clustering, Dendrogram Representation, Density-Based Clustering: DBSCAN and outlier detection, Graph-Based Clustering: Introduction and Applications	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Data Mining Concepts and Techniques J. Han and M. Kamber Morgan Kaufmann, 2006, ISBN 1-55860 901-6
2. Introduction to Data Mining Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education (Addison Wesley), 0-321-32136-7

Suggested Reference Book(s):

1. Mining Massive data sets Anand Rajaram, Jure Leskovec and Jeff Ullman Cambridge University Press 2

Other useful resource(s):

1. https://onlinecourses.nptel.ac.in/noc18_cs14/preview
2. https://www.coursera.org/specializations/data-mining.
3. https://www.futurelearn.com/courses/data-mining-with-weka

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	2	2	3	1	1	1	2	2	2	**2.0**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-5	3	3	3	2	3	1	1	2	3	2	2	**2.3**
Average	**3**	**2.6**	**2.4**	**1.8**	**2.8**	**1**	**1**	**1.2**	**2.2**	**1.8**	**2**	

Data Mining and Data Warehousing Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Relational Database Management Systems (RDBMS), Python Programming

Course Objectives:

1. Practical exposure on implementation of data mining tasks using WEKA and Python Programming by applying well-known standard datasets
2. To learn, understand and implement various data preprocessing techniques
3. Learning performance evaluation on data mining predictive tasks
4. Learning performance evaluation on data mining descriptive tasks
5. Handling a small data mining project for a given practical domain

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Getting familiar with WEKA, ARFF file format, and Python Libraries	Understand
CO2	Apply data preprocessing techniques such as data cleaning, data reduction and data transformation, etc	Apply
CO3	Implementing and evaluating various data mining predictive tasks such as classification and regression	Evaluate
CO4	Implementing and evaluating various data mining descriptive tasks such as clustering and association rule mining	Evaluate
CO5	Create a group project using any of the data mining algorithms and evaluate the performance.	Create

List of Experiments

S. No	Description	Hours
1	To understand WEKA environment and to create and upload ARFF file in WEKA tool.	**2**
2	To implement data cleaning and data reduction techniques by applying various filters in WEKA.	**2**
3	To implement data transformation techniques such as normalization and discretization by applying various filters in WEKA.	**2**
4	• To compute measures of central tendency and measures of data dispersion using the Python NumPy library by loading a dataset in CSV format with the Pandas library. • To implement data visualization using Python Matplotlip library.	**2**
5	To implement various data preprocessing techniques by using Python libraries.	**2**
6	To implement decision trees classification algorithms using WEKA and Python Programming	**2**

7	To implement and compare different ensemble learning algorithms such as Random Forest, AdaBoost, Gradient Boosting, XGBoost, Voting Classifier, and Stacking using WEKA and Python Programming.	**4**
8	To implement and compare partition-based clustering algorithms such as K-Means and K-Medoids using WEKA and Python Programming.	**2**
9	To implement and analyze density-based clustering algorithms such as DBSCAN for cluster formation and outlier detection using WEKA and Python Programming.	**4**
10	To Apriori algorithm for Association Rule Mining using WEKA and Python Programming	**2**
11	Group project based on any of the data mining algorithms.	**4**
	Total Lab hours	**28**

Suggested/Resources:

1. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Pearson Education India, 1st Edition, 2016.
2. J. Han and M. Kamber, Data Mining Concepts and Techniques. Morgan Kaufmann, 3rd Edition, ISBN 978-0-12-381479-1,
3. Witten, Ian H., and Eibe Frank. "Data mining: practical machine learning tools and techniques." Morgan Kaufmann, 2nd Edition, 2012.

Other useful resource(s):

4. Link to NPTEL course contents:https://nptel.ac.in/courses/106105174
5. https://www.youtube.com/watch?v=eYZI7vj5tvE&list=PLH5_eZVldmtV9qLMy4pBYFynNoXl_5YR

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Agile Software Development Process

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Software Engineering

Course Objectives:

1. Understand the principles and practices of agile software development.
2. To study and apply various Agile methodologies and practices such as Scrum, Extreme Programming.
3. To learn and implement Agile design principles, including SOLID principles.
4. To understand Agile testing methodologies, including test planning, test automation, Test-Driven development, user acceptance testing, and continuous integration practices.
5. To develop skills in Agile project management.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the principles and practices of agile software development.	Understand
CO-2	Apply the principles of Agile Software Development, including the Agile Manifesto, to differentiate Agile methods from traditional software development approaches.	Apply
CO-3	Apply agile testing techniques, project management tasks like scheduling, estimation, monitoring and quality assurance activities.	Apply
CO-4	Analyze and implement various Agile methodologies such as Scrum, Extreme Programming (XP), Lean Development, and Feature driven development in software projects.	Analyze
CO-5	Design software systems using Agile design practices and apply SOLID principles to improve code quality, maintainability, and flexibility.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Basics:** The Genesis of Agile, Introduction and background, Agile Manifesto and Principles, Differences between Agile and traditional plans, Stakeholders, Challenges.	4
2	**Agile Approaches-**Extreme Programming, Agile Process Models: Scrum, Project Phases, Dynamic Systems Development Method, Agile Modeling, Agile Unified Process. A Tool Set for the Agile Process, Feature Driven development, Lean Software Development, Agile Project management, Test Driven Development, Continuous Integration, Refactoring, Pair Programming.	8
3	**Agile Design** practices, The Single-Responsibility Principle, The Open-Closed Principle, The Liskov-Substitution Principle, The Dependency-Inversion Principle, The Interface-Segregation Principle.	6

4	**Agile Testing**-Planning and Managing Testing Cycle, Agile Lifecycle and its impact on testing, Principles of Agile Testing, Agile Testing Techniques, xUnit Framework, Test Driven Development, User Acceptance Tests, Test Automation.	**8**
5	**Agile Project Management-** Scheduling in an agile project, scheduling challenges, estimating costs, monitoring.	**2**
Total lectures		**28**

Suggested Text Book(s):

1. Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices , Pearson.
2. James Schiel, Enterprise-Scale Agile Software Development, CRC Press
3. Lisa Crispin, Janet Gregory, "Agile Testing: A Practical Guide for Testers and Agile Teams",
4. Addison Wesley.
5. Roger S. Pressman,Bruce R. Maxim, "Software Engineering", McGraw Hill, 8th edition

Suggested Reference Book(s):

1. Craig Larman, Agile & Iterative development, Pearson.
2. John Hunt, Agile Software construction, Springer.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106105182
2. Link to NPTEL course contents: https://www.youtube.com/watch?v=jRs-aFETAXY
3. Link to topics related to course: https://www.coursera.org/learn/agile-development-and-scrum

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1		2	2	2		3	1	3	1	1	3	**1.6**
CO-2		2	2	2		3			1	2	3	**1.3**
CO-3	2		2	2	1	3	2	1	2	1	3	**1.7**
CO-4	2		2	2	1	3	2	1	2	2	3	**1.8**
CO-5	2		2	2	2	3	2	1	2	3	3	**2**
Average	**2**	**2**	**2**	**2.0**	**0.8**	**3.0**	**1.4**	**1.2**	**1.6**	**1.8**	**3**	

Agile Software Development Process Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: Elective

L-T-P: 0-0-2

Pre-requisite: Object oriented Programming (java is preferred)

Course Objectives:

1. To understand and apply agile methodologies such as Scrum and Extreme Programming (XP) in real-world software development projects.
2. To develop proficiency in modern development **tools** including version control systems, automated build tools, and continuous integration frameworks.
3. To apply agile design and refactoring principles for improving software quality, maintainability, and adaptability.
4. To implement agile project management practices such as backlog creation, user story development, estimation, scheduling, and continuous monitoring.
5. To perform effective testing within agile environments using agile testing techniques to ensure reliable and high-quality software delivery.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Identify the roles and responsibilities in agile projects and their difference from traditional method projects.	Apply
CO2	Apply agile design principles to develop software	Apply
CO3	Employ agile testing techniques to develop software	Apply
CO4	Implement agile project management tasks like scheduling, estimation, monitoring and quality assurance activities	Apply
CO5	Develop software by applying agile design principles and approaches.	Apply

List of Experiments

S. No	Description	Hours
1	**Agile Background & Driving Forces:** To implement and analyze real-world scenarios with frequently changing requirements; compare Agile and Waterfall models; and identify key driving forces such as flexibility, customer collaboration, and faster delivery, using discussion, charts, and presentations.	**4**
2	**Backlog Creation & User Stories:** To implement the learning of Agile practices, select a sample project such as an e-commerce application, create user stories using the standard format, define clear acceptance criteria for each story, and organize them into a prioritized product backlog using tools like Taiga or Jira (Scrum-based).	**2**

3	**Automated Build Tool:** Build tools for compile, package, and manage dependencies in software projects via Maven, Gradle (open source) tools.	**2**
4	**Version Control System:** To implement repository management, commits, branching, and merging via Git, GitHub.	**2**
5	**Continuous Integration (CI):** To implementautomated build and testing, configure a CI pipeline, set it to trigger builds on each code commit, and integrate both build and testing steps using Jenkins.	**2**
6	**Design Principles & Refactoring:** Implement SOLID principles for code smell, refactoring and maintainability and flexibility. Useful tools are Visual Studio Code, Any programming language (Java/Python/C++).	**4**
7	**Agile Testing Practices:** Implement unit testing and basic TDD (test-driven development) with the tools **Junit (Java), Pytest (Python), C++ (**Google Test / Catch2).	**4**
8	**Agile Project Management - Scheduling & Monitoring:** Implement scheduling, estimation, progress monitoring, backlog tracking (burn-down), reporting, and project control in an Agile environment.	**4**
9	**Mini Project (Agile Implementation):** Implement small project environment via scrum, XP by considering all necessary tools and each unit methods.	**4**
	Total Lab hours	**28**

Suggested/Resources:

Suggested Text Book(s):

1. Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices , Pearson.
2. James Schiel, Enterprise-Scale Agile Software Development, CRC Press
3. Lisa Crispin, Janet Gregory, "Agile Testing: A Practical Guide for Testers and Agile Teams", Addison Wesley.
4. Roger S. Pressman,Bruce R. Maxim, "Software Engineering", McGraw Hill, 8th edition

Suggested Reference Book(s):

1. Craig Larman, Agile & Iterative development, Pearson.
2. John Hunt, Agile Software construction, Springer.

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106105182
2. Link to NPTEL course contents: https://www.youtube.com/watch?v=jRs-aFETAXY
3. Link to topics related to course: https://www.coursera.org/learn/agile-development-and-scrum

Evaluation Scheme:

S. No	**Exam**	**Marks**
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	3	2	1	2	3	3	1	2	3	**2.1**
CO-2	2	3	3	3	3	3	3	3	1	3	3	**2.7**
CO-3	2	3	3	3	3	3	3	3	1	3	3	**2.7**
CO-4	2	3	1	2	3	3	3	3	1	3	3	**2.4**
CO-5	2	3	3	3	3	3	3	3	2	3	3	**2.8**
Average	**2.0**	**2.8**	**2.6**	**2.6**	**2.6**	**2.8**	**3.0**	**3.0**	**1.2**	**2.8**	**3.0**	

IoT Analytics

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Programming in Python/C++, Basics of Electronics

Course Objectives:

1. To understand the architecture of IoT systems and the role of hardware platforms such as Arduino Uno, ESP32, and Raspberry Pi in data acquisition.
2. To interface and configure diverse sensors — temperature, humidity, pressure, sound, camera, and light — for real-world IoT data collection.
3. To apply data preprocessing, storage, and communication protocols for reliable IoT data pipelines.
4. To implement statistical and machine learning techniques for analysing, visualising, and deriving insights from IoT sensor data.
5. To design end-to-end IoT analytics solutions with edge processing, cloud integration, and real-time dashboard visualisation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the IoT system architecture and differentiate the capabilities, use-cases, and programming models of Arduino Uno, ESP32, and Raspberry Pi platforms.	Understand
CO-2	Interface and configure multiple sensor types - temperature, humidity, pressure, sound, light, and camera - for IoT data acquisition.	Apply
CO-3	Apply data preprocessing, storage, and communication protocols (MQTT, HTTP) to build reliable IoT data acquisition and transmission pipelines.	Apply
CO-4	Implement statistical analysis, anomaly detection, and machine learning models to extract actionable insights from IoT sensor streams.	Analyse
CO-5	Design and demonstrate an end-to-end IoT analytics system integrating edge processing, cloud connectivity, and real-time data visualisation.	Create

Course Contents:

Unit	Contents	Lectures Required
1	Introduction to IoT Analytics: Definition and scope of IoT analytics — descriptive, diagnostic, predictive, and prescriptive analytics; IoT system layers — sensing, network, data processing, and application; IoT data characteristics — high frequency, heterogeneity, noise, and temporal correlation; Overview of hardware platforms — Arduino Uno, ESP32, and Raspberry Pi: architecture, processing power, memory, connectivity, and use-cases; Selection criteria for hardware platforms and sensors based on application requirements.	3

2	Arduino Uno — ATmega328P architecture, GPIO, ADC/DAC, PWM, I2C, SPI, and UART; Programming with Arduino IDE — sketch structure, libraries, and serial monitor; ESP32 — dual-core architecture, integrated Wi-Fi and Bluetooth, deep-sleep modes, Arduino IDE programming on ESP32; Raspberry Pi — ARM Cortex-A architecture, GPIO, , and Linux-based OS; Python GPIO programming on Raspberry Pi using RPi.GPIO and gpiozero.	4
3	Sensors and Actuators for IoT Analytics: Temperature and humidity sensing —DHT22; ESP32-CAM module (OV2640), ultrasonic sensor (HC-SR04), and IR obstacle sensor; Actuator interfacing — LED, buzzer, relay module, and servo motor;	3
4	IoT Data Acquisition and Communication Protocols: Analog-to-digital conversion (ADC) sampling, resolution, and Nyquist considerations; Serial communication protocols — I2C bus arbitration, SPI full-duplex communication, UART framing; MQTT protocol — broker architecture, topics, QoS levels, and publish-subscribe model; Introduction to IoT platforms — ThingSpeak.	4
5	IoT Data Preprocessing and Storage: Sources of IoT data quality issues — sensor drift, missing readings, noise, and outliers; Techniques for handling missing data — forward-fill, interpolation, and median imputation; Noise filtering — moving average, exponential smoothing, and Butterworth low-pass filter; Feature extraction from time-series sensor data — statistical features (mean, variance, RMS, kurtosis) and sliding window segmentation; Local data storage on Raspberry Pi — CSV logging and SQLite.	5
6	Analytics and Machine Learning for IoT Data: Exploratory data analysis (EDA) — summary statistics, correlation heatmaps, and time-series plots using Pandas and Matplotlib/Seaborn; Statistical methods — Z-score and IQR-based anomaly detection; Threshold-based and rule-based alerting systems; Supervised learning — classification (k-NN, Decision Tree, Random Forest) and regression (Linear Regression, SVR) on sensor data using Scikit-learn; Model evaluation — cross-validation, confusion matrix, RMSE, and MAE; Model persistence and loading using joblib/pickle for edge deployment.	6
7	Edge Analytics and Real-Time Processing: Motivation for edge analytics — latency, bandwidth, and privacy benefits; On-device inference with TensorFlow Lite on Raspberry Pi; Real-time data processing with Python threading and asyncio.	3
	Total lectures	**28**

Suggested Text Book(s):

1. Vijay Madisetti, ArshdeepBahga, Internet of Things: A Hands-On Approach, Universities Press, 2nd Edition, 2021.
2. Simon Monk, Programming Arduino: Getting Started with Sketches, McGraw-Hill, 3rd Edition, 2022.
3. Neil Cameron, ESP32 Arduino Projects, Apress, 2021.
4. Simon Monk, Raspberry Pi Cookbook: Software and Hardware Problems and Solutions, O'Reilly Media, 4th Edition, 2023.

Suggested Reference Book(s):

1. Wes McKinney, Python for Data Analysis, O'Reilly Media, 3rd Edition, 2022.
2. AurélienGéron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.

3. Pete Warden, Daniel Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2020.
4. Guy Hart-Davis, Raspberry Pi Projects for the Evil Genius, McGraw-Hill, 2nd Edition, 2017.

Other useful resource(s):

1. NPTEL course — Internet of Things: https://nptel.ac.in/courses/106105166
2. ESP32 official documentation: https://docs.espressif.com/projects/esp-idf/en/latest/
3. Arduino reference and tutorials: https://www.arduino.cc/reference/en/
4. Raspberry Pi documentation: https://www.raspberrypi.com/documentation/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	1.6
CO-2	3	3	3	2	3	1	1	2	2	2	2	2.2
CO-3	3	3	2	2	3	1	1	1	2	2	2	2.0
CO-4	3	3	3	2	3	1	1	2	3	2	2	2.3
CO-5	3	3	3	3	3	2	2	2	3	2	2	2.5
Average	**3.0**	**2.8**	**2.6**	**2.0**	**2.8**	**1.2**	**1.2**	**1.6**	**2.4**	**1.8**	**2.0**	

IoT Analytics Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming in Python/C++, Basics of Electronics

Course Objectives:

1. To set up and program IoT hardware platforms - Arduino Uno, ESP32, and Raspberry Pi — for sensor-based data acquisition experiments.
2. To interface and experimentally validate multiple sensors including temperature, humidity, pressure, sound, light, and camera modules.
3. To implement IoT communication protocols and transmit sensor data to cloud platforms and local dashboards.
4. To apply data preprocessing, statistical analysis, and machine learning techniques on collected IoT sensor data using Python.
5. To design and demonstrate a complete IoT analytics pipeline integrating hardware sensing, data communication, analysis, and visualisation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Set up development environments and write programs for GPIO, sensor interfacing, and serial communication on Arduino Uno, ESP32, and Raspberry Pi.	Apply
CO-2	Interface and validate the operation of temperature, humidity, pressure, sound, light, and camera sensors on appropriate IoT platforms.	Apply
CO-3	Implement MQTT and HTTP-based IoT data pipelines to transmit sensor readings to cloud platforms and local brokers.	Apply
CO-4	Apply data preprocessing and analysis techniques — cleaning, feature extraction, anomaly detection, and ML-based classification on real sensor datasets.	Analyse
CO-5	Build and demonstrate an integrated, multi-sensor IoT analytics system with real-time data visualisation and automated response logic.	Create

List of Experiments:

S. No	Description	Hours
1	Platform Setup and Blink-to-Buzz: Installation of Arduino IDE, Thonny/MicroPython for ESP32, and Raspberry Pi OS setup. Write and upload programs to blink an LED and activate a buzzer using digital GPIO on Arduino Uno and ESP32. Explore serial monitor for debugging. Verify GPIO, PWM, and timer configurations.	**2**
2	GPIO and Analog Interfacing on Arduino Uno: Interfacing a push button (digital	**2**

	input), LDR (analog input using ADC), and LED (digital output) with Arduino Uno. Read and display analog light levels on the serial monitor. Implement threshold-based LED control responding to ambient light.	
3	Temperature and Humidity Sensing (DHT11/DHT22): Interface DHT11 or DHT22 sensor with Arduino Uno and ESP32. Read temperature and relative humidity values, display them on the serial console, and log timestamped readings to a CSV file via serial. Observe effects of breath and heat source on readings.	**2**
4	Pressure, Altitude, and Environmental Sensing (BMP280/BME680): Interface the BMP280 sensor over I2C with ESP32. Read barometric pressure, temperature, and estimated altitude. Plot the variation of readings with height change (lifting sensor vertically). Use the BME680 (if available) to additionally measure gas/VOC resistance.	**2**
5	Sound and Light Intensity Sensing: Interface a sound sensor module and a BH1750 digital lux sensor with ESP32. Measure ambient sound levels and lux intensity values. Implement event detection — trigger an LED when sound exceeds a threshold. Log both sensor readings simultaneously over time.	**2**
6	Camera Interfacing and Image Capture: Configure the ESP32-CAM module (OV2640) to capture still images and stream MJPEG video over Wi-Fi. Alternatively, use the Raspberry Pi Camera Module v2 to capture images using the picamera2 library in Python. Perform basic image processing operations (grayscale conversion, edge detection) using OpenCV.	**2**
7	MQTT Communication with ESP32: Set up a local MQTT broker (Mosquitto) on Raspberry Pi. Program the ESP32 to publish temperature and humidity data from DHT22 to MQTT topics at fixed intervals. Subscribe to topics from Raspberry Pi using Python (paho-mqtt) and display incoming data on the terminal.	**2**
8	IoT Cloud Integration — ThingSpeak and AWS IoT Core: Publish multi-sensor data (temperature, humidity, lux) from ESP32 to ThingSpeak channels using HTTP GET API. Visualise live data on ThingSpeak dashboards and set up a ThingSpeak React alert. Optionally, connect ESP32 to AWS IoT Core using MQTT with X.509 certificate authentication.	**2**
9	IoT Data Preprocessing and Storage on Raspberry Pi: Collect multi-sensor data (DHT22, BMP280, LDR) from ESP32 via MQTT into a Raspberry Pi Python script. Store data in a CSV file and query it using Pandas. Perform preprocessing — handle missing values, apply moving average smoothing, and visualise time-series trends using Matplotlib and Seaborn.	**2**
10	Anomaly Detection on Sensor Data: Load a pre-collected IoT sensor dataset (temperature, humidity, pressure). Apply Z-score and IQR-based statistical anomaly detection methods to flag abnormal readings. Visualise detected anomalies on time-series plots. Implement a real-time threshold-based alerting mechanism on ESP32 that triggers a buzzer on anomaly.	**2**

11	Machine Learning on IoT Sensor Data: Using Scikit-learn on Raspberry Pi, train a classification model (Decision Tree or Random Forest) on labelled sensor data (e.g., normal vs. fault conditions). Evaluate the model using confusion matrix and F1-score. Save the trained model using joblib and load it for inference on new sensor readings.	**2**
12	Design and demonstrate a complete IoT analytics pipeline: (i) Acquire data from at least three sensors (e.g., DHT22, BMP280, and sound/LDR) using ESP32 ; (ii) Transmit data to Raspberry Pi via MQTT; (iii) Preprocess and visualise data using Python; (iv) Apply a trained ML model for real-time classification; (v) Build a Node-RED dashboard showing live sensor values, alerts, and trends.	**2**
13	Mini Project: End-to-End Multi-Sensor IoT Analytics System.	**4**
Total Lab Hours		**28**

Suggested / Resources:

1. Vijay Madisetti, ArshdeepBahga, Internet of Things: A Hands-On Approach, Universities Press, 2nd Edition, 2021.
2. Simon Monk, Programming Arduino: Getting Started with Sketches, McGraw-Hill, 3rd Edition, 2022.
3. Neil Cameron, ESP32 Arduino Projects, Apress, 2021.
4. Simon Monk, Raspberry Pi Cookbook: Software and Hardware Problems and Solutions, O'Reilly Media, 4th Edition, 2023.
5. Pete Warden, Daniel Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2020.
6. Wes McKinney, Python for Data Analysis, O'Reilly Media, 3rd Edition, 2022.
7. AurélienGéron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.

Other useful resource(s):

1. Arduino reference and tutorials: https://www.arduino.cc/reference/en/
2. ESP32 official documentation: https://docs.espressif.com/projects/esp-idf/en/latest/
3. Raspberry Pi documentation: https://www.raspberrypi.com/documentation/
4. Node-RED documentation: https://nodered.org/docs/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-3	3	3	2	2	3	1	1	1	2	2	2	**2.0**
CO-4	3	3	3	2	3	1	1	2	3	2	2	**2.3**
CO-5	3	3	3	3	3	2	2	2	3	2	2	**2.5**
Average	**3.0**	**2.8**	**2.8**	**2.0**	**3.0**	**1.2**	**1.2**	**1.8**	**2.4**	**1.8**	**2.0**	

Big Data Analytics

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE-2

L-T-P: 2-0-0

Pre-requisite:

The students should have:

1. Basic understanding of Data Structures and Algorithms
2. Fundamental knowledge of Database Management Systems
3. Introductory programming knowledge in Python or Java for data handling
4. Awareness of basic concepts in Operating Systems and Computer Networks (processes, distributed systems, data transfer).

Course Objectives:

1. To understand the fundamentals of Big Data systems, including 5Vs, architecture, and data lifecycle in distributed environments.
2. To comprehend distributed storage and processing frameworks such as Hadoop ecosystem, NoSQL databases, and cloud-based data management systems.
3. To apply modern big data processing techniques using frameworks such as Apache Spark for scalable batch analytics.
4. To analyze real-time and streaming data systems using tools such as Apache Kafka and Apache Flink.
5. To understand scalable analytics techniques for large datasets in distributed environments.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the concepts, architecture, and ecosystem of big data technologies and distributed systems.	Understand
CO-2	Apply distributed storage and processing techniques using Hadoop and Spark frameworks.	Understand, Apply
CO-3	Analyze batch and real-time data processing models using tools like Apache Kafka and Apache Flink.	Apply, Analyze
CO-4	Apply basic machine learning techniques for large-scale data using Spark MLlib.	Analyze
CO-5	Evaluate Big Data architectures and data management strategies for different use cases.	Evaluate
CO-6	Design basic data pipelines integrating storage, processing, and analytics components.	Create

Course Contents:

Unit	Contents	Lectures required

1	**Big Data Fundamentals and Ecosystem:** Evolution of Big Data, 5Vs of Big Data, Traditional vs Big Data Processing, Big Data Architecture Overview, Data Lifecycle and Data Engineering Basics.	3
2	**Distributed Storage and NoSQL Systems:** Hadoop Distributed File System (HDFS), Data Replication and Fault Tolerance, NoSQL databases (Key-value, Document, Column, Graph), CAP theorem, Data consistency models, SQL vs NoSQL systems, Cloud object storage concepts.	4
3	**Distributed Computing and Processing Frameworks:** MapReduce paradigm, Hadoop Ecosystem Overview, Apache Spark Architecture, RDDs, DataFrames, Datasets; DAG Execution Model, In-Memory Processing, Performance Optimization Techniques, GPU-accelerated data processing (overview).	5
4	**Real-Time Data Processing and Streaming Systems:** Batch vs Stream Processing, Stream Data Model, Windowing Techniques, Event Time vs Processing Time, Apache Kafka Concepts, Apache Flink Fundamentals, Lambda and Kappa Architectures, Event-driven data systems	4
5	**Scalable Machine Learning for Big Data**: Introduction to scalable machine learning, Basics of distributed machine learning, Overview of Spark MLlib for classification and regression.	4
6	**Data Engineering and Modern Data Architectures:** Data Warehouse Concepts, Data Lakes and Data Warehouses Architectures, ETL vs ELT, Modern Data Engineering Pipelines, Cloud data engineering workflows	3
7	**Data Ethics, Privacy, and Governance:** Data privacy and governance, Bias and fairness in data systems, Ethical issues in big data analytics, Responsible use of AI in analytics (overview only).	2
8	**Industry Applications and Case Studies:** Smart city analytics systems, Healthcare predictive analytics, Financial fraud detection systems, Cybersecurity anomaly detection systems.	3
	Total lectures	**28**

Suggested Text Book(s):

2. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, First Edition, 2017.
3. Rajaraman & Ullman, Mining of Massive Datasets, Cambridge University Press, Third Edition, 2020.

Suggested Reference Book(s):

4. Bill Chambers &MateiZaharia, Spark: The Definitive Guide, O'Reilly Media, First Edition, 2018
5. Tyler Akidau et al., Streaming Systems, O'Reilly Media, First Edition, 2018.
6. Tom White, Hadoop: The Definitive Guide, O'Reilly Media, Fourth Edition, 2015.

Other useful resource(s):

1. Coursera, Big Data Specialization Courses, Available at:

https://www.coursera.org/specializations/big-data
2. edX, Data Engineering and Analytics Programs, Available at: https://www.edx.org/learn/data-engineering
3. AWS Training and Certification (Big Data & Analytics), Available at: https://aws.amazon.com/training/learn-about/big-data/
4. Google Cloud Big Data and Machine Learning Learning Path, Available at: https://cloud.google.com/training/data-ml
5. Apache Spark Official Documentation, Available at: https://spark.apache.org/docs/latest/
6. Apache Kafka Official Documentation, Available at: https://kafka.apache.org/documentation/
7. Apache Flink Official Documentation, Available at: https://flink.apache.org/learn-flink/

Evaluation Scheme:

S. No.	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Big Data Analytics Lab

COURSE CODE: XXXXXXXX
COURSE CREDITS: 1
CORE/ELECTIVE: CORE
L-T-P: 0-0-2

Pre-requisite: Big Data Analytics (Theory), Python/Java Programming, Basic SQL

Course Objectives:

1. To implement distributed data processing concepts using big data frameworks.
2. To apply data ingestion, storage, and processing techniques using Hadoop ecosystem tools.
3. To develop batch data processing workflows using Apache Spark.
4. To analyze real-time streaming data using Apache Kafka-based pipelines.
5. To implement basic machine learning workflows on large datasets using Spark MLlib.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Configure and execute big data processing environments using Hadoop ecosystem tools.	Apply
CO2	Implement distributed data processing workflows using Spark for batch analytics.	Apply
CO3	Develop real-time data streaming applications using Kafka-based pipelines.	Apply
CO4	Apply machine learning techniques on large datasets using Spark MLlib.	Apply
CO5	Design and implement a simple end-to-end big data pipeline integrating data ingestion, batch processing, streaming, and analytics.	Apply

List of Experiments

S. No.	Description	Hours
1.	Installation and configuration of Hadoop ecosystem; HDFS commands and file operations; data upload and retrieval.	**4**
2.	Implementation of MapReduce programs for basic data processing tasks (word count, frequency analysis, sorting).	**4**
3.	Spark environment setup; RDD operations and DataFrame-based data processing.	**4**
4.	Implementation of batch analytics using Spark SQL and data transformation pipelines.	**4**
5.	Real-time data streaming using Apache Kafka: producer-consumer model and message processing.	**4**
6.	Stream processing using Apache Kafka integrated with Spark Structured Streaming for real-time analytics.	**4**

7.	Machine learning using Spark MLlib (classification/regression on large datasets).	**2**
8.	Mini project: Design and implement an integrated big data pipeline including data ingestion, batch processing, streaming, and visualization using Spark and Hadoop ecosystem tools.	**2**
	Total Lab hours	**28**

Suggested/Resources:

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly, 2017.
2. Tom White, Hadoop: The Definitive Guide, O'Reilly, 4th Edition, 2015.
3. Bill Chambers, MateiZaharia, Spark: The Definitive Guide, O'Reilly, 2018.
4. Tyler Akidau et al., Streaming Systems, O'Reilly, 2018.
5. Rajaraman & Ullman, Mining of Massive Datasets, Cambridge University Press, 3rd Edition, 2020.

Other useful resource(s):

1. NPTEL Courses on Big Data and Hadoop: https://nptel.ac.in
2. Coursera Big Data Specialization: https://www.coursera.org/specializations/big-data
3. AWS Big Data Learning: https://aws.amazon.com/training/learn-about/big-data/
4. Google Cloud Data Engineering: https://cloud.google.com/training/data-ml
5. Official Documentation:
   i) https://spark.apache.org/docs/latest/
   ii) https://kafka.apache.org/documentation/
   iii) https://hadoop.apache.org/
   iv) https://flink.apache.org/

Evaluation Scheme:

S. No.	Exam	Marks
1.	Mid Sem. Evaluation (P-1)	20
2.	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Search in AI

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective-2

L-T-P: 2-0-0

Pre-requisite: Data Structure, Discrete Structure

Course Objectives:

1. Describe introductory techniques in Artificial Intelligence
2. Heuristic search and adversarial search, Logic for knowledge representation and reasoning
3. Reasoning under uncertainty
4. Apply introductory techniques in Artificial Intelligence to solve realistic problems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understanding and apply blind/uninformed algorithms in the state space search	Understand, Apply
CO-2	Understanding and apply efficient heuristic search and hill-climbing algorithms	Understand, Apply
CO-3	Understanding and apply genetic algorithms	Understand, Apply
CO-4	Understanding and apply optimal path algorithms	Understand, Apply
CO-5	Understanding and apply algorithms for game theory	Understand, Apply

Course Contents:

Unit	Contents	Lectures required
1	Introduction: Intelligence, Intelligent Agents, State Space Search, Domain Independent Algorithms, Deterministic Search, Breadth-first, Depth-first and Depth First Iterative Deepening Search,	**5**
2	Heuristic Search, Hill-Climbing Algorithm, Heuristic Functions and the Search Landscape, Solution Space Search, The Travelling Salesman Problem (TSP), Escaping Local Optima, Stochastic Local Search	**6**
3	Genetic Algorithms (GA), Satisfiability Problem (SAT), GA for the TSP, Emergent Systems, and Ant Colony Optimization	**5**
4	Finding Optimal Paths, Branch & Bound, Dijkstra’s Algorithm, Algorithm A*, and Admissibility of A*	**6**
5	Game Theory, Popular Recreational Games, Board Games and Game Trees, The Evaluation Function in Board Games, Algorithm Minimax and Alpha-Beta Pruning	**6**

Total lectures	28

Suggested Text Book(s):

1. Artificial Intelligence a Modern Approach, 3rd Edition.
2. Prentice Hal 2. Artificial Intelligence Hardcover by Elaine Rich and Kevin Knight Approved in Academic Council held on 28 July 2021

Suggested Reference Book(s):

1. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp by Peter Norvig
2. Machine Learning by Tom M. Mitchell
3. Prediction Machines: The Simple Economics of Artificial Intelligence by Ajay Agrawal , Joshua Gans , Avi Goldfarb

Other useful resource(s):

1. Link to NPTEL course contents:
   i. https://nptel.ac.in/courses/106105077/
   ii. https://nptel.ac.in/courses/106105079/
2. Link to topics related to course:
   i. https://www.ibm.com/developerworks/library/cc-beginner-guide-machine-learning-ai-cognitive/index.html
   ii. https://ai.google/education/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Search in AI Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: Elective-2

L-T-P: 0-0-2

Pre-requisite: Data Structure level programming in C++/Python

Course Objectives:

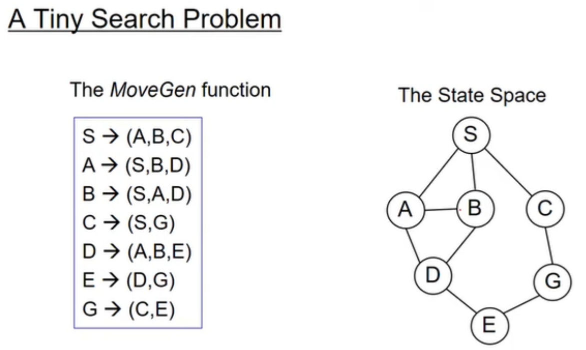
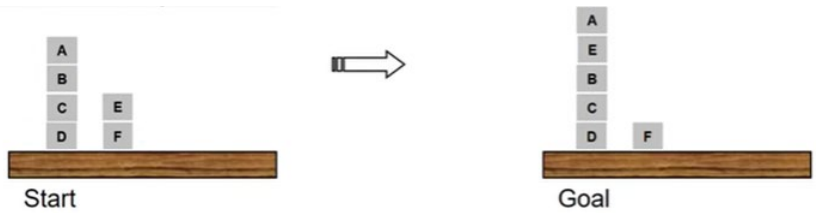
1. To understand and implement object-oriented concepts such as classes, constructors, destructors, and friend functions.
2. To apply encapsulation, inheritance, polymorphism, and abstraction in solving real-world programming problems.
3. To utilize the Standard Template Library (STL) for writing efficient and reusable code.
4. To develop robust programs using exception handling techniques.
5. To implement linear data structures and recursive algorithms using dynamic memory and STL.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Applying blind/uninformed algorithms in the state space search	Apply
CO-2	Applying efficient heuristic search and hill-climbing algorithms	Apply
CO-3	Applying genetic algorithms	Apply
CO-4	Applying optimal path algorithms	Apply
CO-5	Applying algorithms for game theory	Apply

List of Experiments

S. No	Description	Hours
1	Implement state space search program for the following problems– The Water Jug Problem, The Eight Puzzle Problem, The Man-Goat-Lion-Cabbage Problem, and the Six Queens Problem	**4**
2	Implement state space search algorithms using MoveGen function and tracing the path for the problem given below.	**2**

		
3	Implement deterministic Depth First Search. Breadth First Search, and Depth First Search Iterative deepening state space search algorithms for the graph in # problem 2 above.	**4**
4	Implement Heuristic Search Algorithm for (a) the 8 Puzzle problem (b) Solve the following Block World Problem shown in the Fig. with a given start state and a goal state using Hill-Climbing Search Algorithm having the heuristic function h = Add n if the block is on a correct structure of n blocks; Subtract n if the block is on a wrong structure of n blocks Given h(Start) = -1 and h(Goal) = 10 	**2**
5	Implement Hill-Climbing algorithm for a given Travelling Salesman problem.	**2**
6	Implement Stochastic Local Search- stochastic hill-climbing algorithm.	**2**
7	Implement Genetic algorithm for the following SAT problem and for the TSP problem. $(b \vee \sim c) \wedge (c \vee \sim d) \wedge (\sim b) \wedge (\sim a \vee \sim e) \wedge (e \vee \sim c) \wedge (\sim c \vee \sim d)$	**2**
8	Implement an Ant Colony Optimization algorithm.	**2**
9	Implement optimal paths finding using Branch and Bound and Dijkstra Algorithm	**2**
10	Implement an optimal path finding using A* Algorithm	**2**
11	Implement Board Games and Game Trees using Algorithm Minimax	**2**
12	Implement Board Games and Game Trees using Alpha-Beta Pruning	**2**
13	Project Work	Whole Semester
	Total Lab hours	**28**

Suggested Text Book(s):

1. Artificial Intelligence a Modern Approach, 3rd Edition.
2. Prentice Hal 2. Artificial Intelligence Hardcover by Elaine Rich and Kevin Knight Approved in Academic Council held on 28 July 2021

Suggested Reference Book(s):

1. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp by Peter Norvig
2. Machine Learning by Tom M. Mitchell
3. Prediction Machines: The Simple Economics of Artificial Intelligence by Ajay Agrawal , Joshua Gans , Avi Goldfarb

Other useful resource(s):

1. Link to NPTEL course contents:
   i. https://nptel.ac.in/courses/106105077/
   ii. https://nptel.ac.in/courses/106105079/
2. Link to topics related to course:
   i. https://www.ibm.com/developerworks/library/cc-beginner-guide-machine-learning-ai-cognitive/index.html
   ii. https://ai.google/education/

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	3	1	3	1	1	2	2	1	2	**1.9**
CO-2	3	3	3	1	3	1	1	2	2	1	2	**2.0**
CO-3	3	2	2	1	3	1	1	1	1	1	2	**1.6**
CO-4	3	2	2	2	2	1	1	1	1	1	2	**1.6**
CO-5	3	2	3	2	3	1	1	1	2	2	2	**2.0**
Average	**3.0**	**2.2**	**2.6**	**1.4**	**2.8**	**1.0**	**1.0**	**1.4**	**1.6**	**1.2**	**2.0**	

Intelligent Robotics and Autonomous Systems

COURSE CODE:

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Course Objectives:

1. To understand the structure and communication mechanisms of ROS2 for robotic system development.
2. To model robotic systems and apply sensing, localization, and planning concepts for autonomous operation.
3. To integrate AI-driven perception and decision-making techniques in autonomous robotic platforms.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Explain the structure and components of ROS2 and their interaction.	Understanding
CO2	Model robotic systems and analyze kinematics for wheeled and quadruped platforms.	Analyzing
CO3	Describe the role of sensors and perception in localization and mapping.	Understanding
CO4	Apply motion planning and control strategies for robot navigation and locomotion.	Applying
CO5	Evaluate the use of AI in autonomous decision-making and robot perception.	Evaluating

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Robotics and ROS Framework** Overview of mobile, legged, and humanoid robots; ROS2 architecture and communication (nodes, topics, services, actions, parameters); DDS middleware overview; introduction to simulation tools (RViz2, Gazebo, rqt, rosbag).	4
2	**Robot Modeling, Simulation, and Kinematics** URDF and Xacro fundamentals; coordinate frames and transformations (tf2); forward and inverse kinematics for wheeled robots (differential drive, mecanum); overview of quadruped kinematics; simulation workflow and integration with Gazebo.	6
3	**Sensing, Perception, and Localization** Principles of sensing (IMU, depth cameras, LiDAR); perception pipeline for visual and depth data; sensor calibration and ROS2 sensor drivers; localization concepts (odometry, EKF/UKF overview); SLAM techniques (2D SLAM, Visual SLAM) and ROS2 SLAM toolbox.	5

4	**Path Planning and Control Strategies** Overview of path planning algorithms (Dijkstra, A*, RRT); behavior trees for navigation; ROS2 Navigation Stack (Nav2) architecture; trajectory tracking and feedback control (PID, Pure Pursuit); MPC overview for robotic control; stability and balance concepts for legged robots.	6
5	**Artificial Intelligence and Autonomous Systems** AI-based perception and decision-making; deep learning for robot vision; introduction to reinforcement learning for navigation and control; edge AI with NVIDIA Jetson; AI in quadruped locomotion; integration of AI with ROS2; case studies — Unitree Go2 quadruped and AI-based autonomous vehicle.	7
Total lectures		**28**

Suggested Text Book(s):

1. B. Siciliano and L. Sciavicco, Robotics: Modelling, Planning and Control, 1st ed., Springer, 2009.
2. Francisco Martín Rico, A Concise Introduction to Robot Programming with ROS 2, 2nd ed., Chapman & Hall, 2025.

Suggested Reference Book(s):

1. Marco Matteo Bassa, A very informal journey through ROS 2: patterns, anti-patterns, frameworks and best practices, Leanpub, 2023.
2. Myles and C. Halesworth, Path Planning with ROS2 Algorithms for Autonomous Navigation: Optimizing A and RRT for Robot Motion in Gazebo Environments*, Kindle ed., Independently Published, 2025.

Other useful resource(s):

1. ETH Zürich, Programming for Robotics (ROS2), [Online Course Material], Available: https://rsl.ethz.ch/education-students/lectures/ros.html
2. Murilo M. Marinho, ROS2 Tutorial [Online Course Material], Available: https://ros2-tutorial.readthedocs.io/_/downloads/en/latest/pdf/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered up to T-1
2	T-2	25	1.5 Hours	Syllabus covered up to T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignments (2): 10 Marks Quizzes (2) : 10 Marks Attendance : 5 Marks

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

CO/PO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	Average
CO1	3	2	2	2	3	1	1	1	1	1	2	**1.7**
CO2	3	3	3	3	3	1	1	1	1	1	2	**2.0**
CO3	3	3	2	3	3	1	1	1	1	1	2	**1.9**
CO4	3	3	3	3	3	1	1	1	1	2	2	**2.1**
CO5	3	3	3	3	3	1	1	1	1	2	3	**2.2**
Average	**3.0**	**2.8**	**2.6**	**2.8**	**3.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.4**	**2.2**	**1.9**

Intelligent Robotics and Autonomous Systems Lab

COURSE CODE:

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Course Objectives:

1. To develop proficiency in ROS2 tools, node communication, and launch workflows for robotics experiments.
2. To model, simulate, and interface robotic platforms with perception sensors in Gazebo and RViz.
3. To implement and evaluate navigation, control, and AI-enabled autonomy on wheeled and legged robots.

Course outcomes:

S. No	Course outcomes	Level of Attainment
CO1	Configure and operate ROS2 environments, develop and debug nodes, and perform communication between multiple nodes.	Understanding
CO2	Model, simulate, and analyze robotic systems using Gazebo and RViz.	Applying
CO3	Interface, calibrate, and process data from sensors (IMU, Depth Camera, LIDAR) in ROS2.	Analyzing
CO4	Implement and evaluate perception and control algorithms for wheeled and legged robots.	Evaluating
CO5	Integrate AI-based decision-making and demonstrate autonomous operations.	Creating

List of Experiments:

S. No	Description	Hours
1	**Understanding ROS2 and its ecosystem** Introduction to ROS2 workspace, nodes, topics, parameters, and basic CLI tools.	2
2	**Publisher–Subscriber Communication** Write ROS2 publisher and subscriber nodes; visualize message flow using rqt_graph.	2
3	**Services and Actions in ROS2** Create service server/client pairs and implement action servers for long-running tasks.	2
4	**ROS2 Parameters and Launch Files** Explore dynamic parameters; create launch files to run multi-node systems.	2
5	**Robot Modeling and Simulation** Model a differential drive or quadruped robot using URDF/Xacro and simulate in Gazebo &RViz.	2
6	**Sensor Integration** Interface Depth Camera and LiDAR; acquire and publish sensor data; visualize data in RViz; perform *basic* point cloud processing.	2
7	**Teleoperation and Motor Control**Control Autonomous AI Car using ROS2; implement basic teleoperation and velocity control.	2
8	**Lane Detection and Following** Implement computer vision-based lane detection and integrate with motion control.	2
9	**SLAM and Navigation** Perform mapping and localization using ROS2 SLAM Toolbox; implement autonomous navigation with obstacle avoidance using Nav2.	2
10	**Unitree Go2 Robot Control** Connect to Unitree SDK; execute predefined gaits; monitor joint and state data in RViz.	2
11	**Vision-based Object Tracking** Use camera feeds to detect and track	2

	objects; integrate tracking with robot motion.	
12	**AI-based Obstacle Avoidance** Deploy AI models (YOLO/ResNet) on NVIDIA Jetson; integrate perception with control for autonomous navigation.	**2**
Mini Project	Integrate sensing, perception, planning, and control to demonstrate end-to-end autonomous behavior on selected platform.	
	Total Lab Hours	**24**

Suggested/Resources:

1. ROS2 Official Tutorials, [Online Documentation], Available: https://docs.ros.org/en/foxy/Tutorials.html
2. Open Source Robotics Foundation, Gazebo and RViz Documentation, [Online], Available: https://gazebosim.org/docs/latest/
3. ETH Zürich, Programming for Robotics (ROS2), [Online Course Material], Available: https://rsl.ethz.ch/education-students/lectures/ros.html
4. Unitree Robotics, Unitree Go2 SDK and ROS2 Integration Guide, Available: https://www.unitree.com/opensource

Other useful resource(s):

5. NVIDIA, Jetson Developer Resources (JetPack SDK), Available: https://developer.nvidia.com/embedded/jetpack

Evaluation Scheme:

S. No.	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs):

CO/PO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	Average
CO1	2	2	1	1	3	1	1	1	2	1	2	**1.5**
CO2	3	2	2	2	3	1	1	2	1	1	2	**1.8**
CO3	3	3	2	3	3	1	1	1	1	1	2	**1.9**
CO4	3	3	3	3	3	1	1	2	2	2	2	**2.3**
CO5	3	3	3	3	3	1	1	2	2	2	3	**2.4**
Average	**2.8**	**2.6**	**2.2**	**2.4**	**3.0**	**1.0**	**1.0**	**1.6**	**1.6**	**2.2**	**2.2**	**2.0**

Image Processing and Computer Vision

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Linear Algebra, Probability, Machine Learning and Deep learning

Course Objectives:

1. To understand the fundamentals of digital image processing and computer vision.
2. To learn image representation, enhancement, and transformation techniques.
3. To apply feature extraction and segmentation techniques for image analysis.
4. To understand machine learning and deep learning approaches in computer vision.
5. To design and implement real-world vision-based applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain fundamental concepts of image processing and computer vision	Understand
CO-2	Apply image enhancement, filtering, and transformation techniques.	Apply
CO-3	Analyze images using segmentation and feature extraction methods	Apply
CO-4	Apply machine learning and deep learning techniques for vision tasks.	Analyze
CO-5	Design and implement real-world computer vision applications.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction and Image Formation:** Introduction and goals of computer vision and image processing; applications in autonomous driving, medical imaging, and robotics; image formation concepts; radiometry basics; geometric transformations; geometric camera models: pinhole model; sampling and quantization; camera calibration; image formation in stereo vision.	**6**

2	**Image Processing Techniques:** Image representation: spatial and intensity resolution; color models: RGB, HSV; intensity transformations and histogram equalization; noise models: Gaussian, salt-and-pepper noise; spatial filtering: Gaussian, median, bilateral; edge detection: Sobel; multi-scale representation image pyramids Gaussian and Laplacian; basics of Image restoration noise removal, de-blurring, and image degradation model.	**7**
3	**Image Segmentation and Feature Extraction:** Image segmentation techniques: thresholding (Otsu's method), region-based methods, clustering-based segmentation, watershed algorithm; segmentation evaluation metrics: IoU, Dice coefficient; feature detection: edges, Harris corner detector; feature descriptors: SIFT, SURF, ORB, HOG; feature matching and outlier removal: feature matching, RANSAC.	**5**
4	**Machine Learning for Computer Vision:** Supervised and unsupervised learning; linear and logistic regression; K-nearest neighbors (KNN); support vector machines (SVM); dimensionality reduction (PCA, LDA)	**5**
5	**Deep Learning for Computer Vision:** Neural network fundamentals; convolutional neural networks (CNNs); architectures (LeNet, AlexNet, VGG, ResNet); image classification; introduction to object detection (R-CNN, Fast R-CNN, YOLO); semantic segmentation (FCN, U-Net basics); transfer learning and fine-tuning.	**5**
Total lectures		**28**

Suggested Text Book(s):

1. Computer Vision and Image Processing: Fundamentals and Applications, "Computer Vision and Image Processing: Fundamentals and Applications", CRC Press, USA.
2. Computer Vision: A Modern Approach, "Computer Vision - A Modern Approach", Pearson Education.
3. Computer Vision: Algorithms and Applications, "Computer Vision - Algorithms & Applications", Springer.

Suggested Reference Book(s):

1. Digital Image Processing, "Digital Image Processing", Rafael C. Gonzalez and Richard E. Woods, Pearson.
2. Deep Learning, "Deep Learning", Ian Goodfellow, YoshuaBengio, and Aaron Courville, MIT Press.
3. Learning OpenCV, "Learning OpenCV", Gary Bradski and Adrian Kaehler, O'Reilly Media.

Other useful resource(s):

4. Link to NPTEL course contents: https://nptel.ac.in/courses/108103174
5. Link to NPTEL course contents: https://nptel.ac.in/courses/117105135

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	1	1	2	**1.4**
CO-2	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-3	3	3	3	2	3	1	1	1	1	1	2	**1.9**
CO-4	3	3	2	2	3	1	1	1	1	1	3	**1.9**
CO-5	3	2	3	2	3	1	1	2	2	2	3	**2.1**
Average	**3**	**2.6**	**2.2**	**1.8**	**2.8**	**1**	**1**	**1.2**	**1.2**	**1.2**	**2.4**	

Image Processing and Computer Vision Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming (Python/C++), and fundamentals of Linear Algebra and Probability.

Course Objectives:

1. To introduce the fundamental concepts of digital image processing and computer vision.
2. To provide knowledge of image representation, enhancement, and transformation techniques.
3. To develop understanding of image segmentation and feature extraction methods for visual data analysis.
4. To familiarize students with machine learning techniques applied to computer vision problems.
5. To expose students to deep learning approaches for advanced vision tasks such as image classification, object detection, and segmentation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Understand the fundamentals of digital image processing including image representation, sampling, quantization, and geometric transformations.	Apply
CO2	Analyze and apply image enhancement and filtering techniques for improving image quality and noise removal.	Apply
CO3	Analyze image data using segmentation and feature extraction techniques for meaningful interpretation.	Apply
CO4	Apply machine learning algorithms for image classification and evaluate model performance using appropriate metrics.	Apply
CO5	Design and implement deep learning models for computer vision tasks such as image classification, object detection, and segmentation.	Apply

List of Experiments

S. No	Description	Hours
1	Write a program to study image representation, sampling, and quantization of images.	2
2	Write a program to perform geometric transformations such as translation, rotation, and scaling on images.	2
3	Write a program to implement image enhancement techniques including histogram equalization and intensity transformations.	2

4	Write a program to add noise (Gaussian and salt & pepper) to images and perform noise removal using spatial filters such as Gaussian, median, and bilateral filters.	2
5	Write a program to perform edge detection using Sobel, Laplacian, and Canny operators.	2
6	Write a program to implement frequency domain filtering using Fourier Transform, including low-pass and high-pass filters.	2
7	Write a program to implement image segmentation techniques such as Otsu thresholding, K-means clustering, and watershed algorithm.	2
8	Write a program to perform feature extraction using methods such as Harris corner detection, SIFT/ORB descriptors, and HOG features	2
9	Write a program to implement image classification using K-Nearest Neighbors (KNN) with dimensionality reduction using PCA, and evaluate the results using precision, recall, F1-score, and IoU metrics.	2
10	Write a program to implement image classification using Support Vector Machine (SVM) with dimensionality reduction using PCA, and evaluate the performance using precision, recall, F1-score, and IoU metrics.	2
11	Write a program to implement CNN-based image classification for computer vision tasks.	2
12	Write a program to perform image classification using transfer learning with pre-trained deep learning models.	2
13	Write a program to implement basic object detection using YOLO or semantic segmentation using U-Net for computer vision applications.	4
Total Lab hours		**28**

Suggested Text Book(s):

1. Computer Vision and Image Processing: Fundamentals and Applications, “Computer Vision and Image Processing: Fundamentals and Applications”, CRC Press, USA.
2. Computer Vision: A Modern Approach, “Computer Vision - A Modern Approach”, Pearson Education.
3. Computer Vision: Algorithms and Applications, “Computer Vision - Algorithms & Applications”, Springer.

Suggested Reference Book(s):

1. Digital Image Processing, “Digital Image Processing”, Rafael C. Gonzalez and Richard E. Woods, Pearson.
2. Deep Learning, “Deep Learning”, Ian Goodfellow, YoshuaBengio, and Aaron Courville, MIT Press.
3. Learning OpenCV, “Learning OpenCV”, Gary Bradski and Adrian Kaehler, O’Reilly Media.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/108103174
2. Link to NPTEL course contents: https://nptel.ac.in/courses/117105135

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	3	1	1	1	2	1	2	**1.7**
CO-2	3	3	2	2	3	1	1	1	2	1	2	**1.9**
CO-3	3	3	3	2	3	1	1	1	1	1	2	**1.9**
CO-4	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-5	3	2	3	2	3	1	1	2	2	2	3	**2.1**
Average	**3**	**2.6**	**2.4**	**1.8**	**3**	**1**	**1**	**1.2**	**1.6**	**1.2**	**2.2**	

Introduction to BlockChain Technology

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Basics of Computer Networks, Fundamentals of Programming

Course Objectives:

1. To understand the fundamental concepts of blockchain technology, including its history, architecture, and types.
2. To explain cryptographic primitives (hashing, digital signatures) and their role in ensuring blockchain security.
3. To analyze the working of consensus algorithms (PoW, PoS, etc.) and their impact on sustainability and decentralization.
4. To explore smart contracts, decentralized applications (dApps), and blockchain platforms like Ethereum.
5. To evaluate real-world applications of blockchain beyond cryptocurrencies (e.g., supply chain, healthcare, voting).

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts, architecture, and types of blockchain (public, private, consortium).	Understand
CO-2	Apply cryptographic techniques (hash functions, digital signatures) to ensure data integrity and non-repudiation in blockchain.	Apply
CO-3	Analyze different consensus mechanisms (Proof of Work, Proof of Stake, etc.) for their strengths, weaknesses, and energy implications.	Analyze
CO-4	Design basic smart contracts and explain the working of decentralized applications (dApps) on platforms like Ethereum.	Create
CO-5	Evaluate the societal, legal, and environmental impact of blockchain adoption across various industry sectors.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Blockchain:** Evolution of digital trust, centralized vs. decentralized systems, history of blockchain (Bitcoin whitepaper), basic terminology (blocks, chain, nodes, miners), types of blockchain (public, private, consortium, hybrid).	**4**
2	**Cryptographic Foundations:** Hash functions (SHA-256, properties), Merkle trees, digital signatures (ECDSA), public-private key pairs, wallets and addresses, immutability and data integrity.	**5**

3	**Blockchain Architecture & Transactions:** Structure of a block (header, body), block hash, previous block hash, nonce, transaction lifecycle (creation, broadcast, validation), UTXO vs. account-based models.	4
4	**Consensus Mechanisms:** Need for consensus, Byzantine Generals' Problem, Proof of Work (PoW) – mining, difficulty, energy concerns, Proof of Stake (PoS), Delegated PoS, Practical Byzantine Fault Tolerance (PBFT), and other eco-friendly alternatives.	5
5	**Smart Contracts & Ethereum:** Introduction to Ethereum, Ether and gas, structure of a smart contract, Solidity basics (simple contract deployment), Decentralized Applications (dApps), oracles.	5
6	**Blockchain Use Cases & Challenges:** Beyond cryptocurrencies – supply chain management, digital identity, healthcare records, voting systems, intellectual property. Scalability, privacy (zero-knowledge proofs), interoperability, legal and regulatory aspects.	5
Total lectures		**28**

Suggested Text Book(s):

1. ArshdeepBahga, Vijay Madisetti, *Blockchain Applications: A Hands-On Approach*, VPT, 2017.
2. Imran Bashir, *Mastering Blockchain*, Packt Publishing, 3rd Edition, 2020.
3. Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media, 2nd Edition, 2017.

Suggested Reference Book(s):

1. Daniel Drescher, Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress, 2017.
2. William Mougayar, *The Business Blockchain*, Wiley, 2016.

Other useful resource(s):

1. NPTEL: Blockchain Architecture Design and Use Cases – https://nptel.ac.in/courses/106105235
2. MIT OpenCourseWare: Blockchain and Money – https://ocw.mit.edu (15.S12)
3. Solidity Documentation – https://docs.soliditylang.org

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	2	1	1	2	1	2	**1.6**
CO-2	3	3	2	2	3	1	2	1	1	1	2	**1.9**
CO-3	2	3	2	3	2	3	1	1	1	1	3	**2.0**
CO-4	3	2	3	2	3	2	1	2	2	2	3	**2.3**
CO-5	2	3	2	2	1	3	3	2	3	3	3	**2.5**
Average	**2.6**	**2.6**	**2.0**	**2.0**	**2.2**	**2.2**	**1.6**	**1.4**	**1.8**	**1.6**	**2.6**	

Introduction To Blockchain Technology Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Basics of Computer Networks, Fundamentals of Programming, Linux/Command Line Basics

Course Objectives:

1. To install and configure blockchain development environments (Ganache, Truffle, MetaMask).
2. To implement cryptographic primitives (hashing, digital signatures) using Python/JavaScript libraries.
3. To simulate and analyze different consensus mechanisms (PoW, PoS) through hands-on coding.
4. To develop, deploy, and test smart contracts using Solidity and Ethereum platforms.
5. To build a basic decentralized application (dApp) and evaluate blockchain use cases.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Install and configure blockchain development tools and create a local blockchain network.	Apply
CO2	Implement cryptographic hash functions and digital signatures for data integrity and authentication.	Apply
CO3	Simulate and compare consensus algorithms (Proof of Work, Proof of Stake) through programming.	Analyze
CO4	Write, deploy, and test smart contracts on Ethereum-compatible blockchain using Solidity and Remix IDE.	Create
CO5	Develop a simple decentralized application (dApp) integrating smart contract with a front-end interface.	Create

List of Experiments

S. No	Description	Hours
1	Installation and setup of blockchain development environment: Node.js, npm, Ganache (personal blockchain), Truffle framework, MetaMask browser extension. Creating a local blockchain network with multiple test accounts.	2
2	Implementation of cryptographic hash functions (SHA-256) using Python/JavaScript. Understanding properties: deterministic, pre-image resistance, avalanche effect. Hashing of text, files, and blocks.	2
3	Implementation of digital signatures using ECDSA (Elliptic Curve Digital Signature Algorithm): key pair generation, signing a message, and signature verification. Simulating transaction authentication.	2
4	Building a simple blockchain in Python/JavaScript: Block structure (index, timestamp, data, previous hash, nonce, current hash), proof-of-work, chain validation, and tamper detection.	2

5	Simulation of Proof of Work (PoW) consensus: Mining with difficulty adjustment, computing nonce, measuring mining time and energy cost. Comparison with Proof of Stake (PoS) simulation (validator selection based on stake).	**2**
6	Design public-private key pairs and create Ethereum addresses, send signed transactions using Web3.js library. Also check balances on Ganache.	**2**
7	Write a basic Solidity smart contract (storage, counter, simple bank). Compile and deploy to local Ganache blockchain, and interact using Remix interface.	**2**
8	Implement Smart contract development – Variables, data types, functions, modifiers, and events in Solidity. Deploy a contract for transferring Ether between accounts.	**2**
9	Implement a simple token contract (ERC-20 like) with minting and transferring functions.	**2**
10	Interact with smart contracts using Web3.js: Connect to Ganache, read contract data, send transactions from different accounts, and listen to events.	**2**
11	Developing a decentralized application (dApp) – Part 1: Setting up project structure (HTML/CSS/JS front-end), connecting MetaMask, displaying account and network details.	**4**
12	Developing a decentralized application (dApp) – Part 2: Integrating a deployed smart contract with front-end, calling read/write functions, handling transaction confirmations.	**4**
	Total Lab hours	**28**

Suggested/Resources:

3. Andreas M. Antonopoulos, Mastering Ethereum, O'Reilly Media, 2018.
4. Imran Bashir, Mastering Blockchain, Packt Publishing, 3rd Edition, 2020.
5. Dapp University (YouTube & GitHub) – Smart contract and dApp tutorials.
6. Solidity Documentation: https://docs.soliditylang.org
7. Web3.js Documentation: https://web3js.readthedocs.io
8. Truffle Suite Documentation: https://trufflesuite.com/docs
9. Remix IDE: https://remix.ethereum.org
10. CryptoZombies (Interactive Solidity tutorial): https://cryptozombies.io

Other useful resource(s):

1. NPTEL: Blockchain Architecture Design and Use Cases – https://nptel.ac.in/courses/106105235
2. MetaMask Documentation: https://docs.metamask.io
3. Etherscan (Blockchain Explorer): https://etherscan.io

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	1	1	3	1	1	2	2	1	2	**1.6**
CO-2	3	2	2	2	3	1	2	1	1	1	2	**1.8**
CO-3	2	3	2	3	2	2	1	1	1	1	3	**1.9**
CO-4	2	2	3	2	3	1	1	2	2	2	3	**2.1**
CO-5	2	2	3	2	3	2	2	3	3	2	3	**2.5**
Average	**2.2**	**2.2**	**2.2**	**2.0**	**2.8**	**1.4**	**1.4**	**1.8**	**1.8**	**1.4**	**2.6**	

Computing for Data Science

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: Python Programming

Course Objectives:

1. To understand the concept of Data Science Process, Types & Levels of data, Data Munging, Feature Engineering.
2. To learn Statistical Methods in Data Science.
3. To implement Technologies & Tools in Database Analytics.
4. To analyze and debug code with Computing techniques for Data Science.
5. To analyze and evaluate the Statistical Methods for Evaluation and Exploratory Data Analysis & Data Science Process-.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Make use of basic concepts, methods, and mathematics relevant to computational techniques for data science	Apply
CO-2	Develop own statistical analyses and implement them with advanced statistical programming tools	Apply
CO-3	Develop and apply advanced and associated computing techniques and technologies.	Apply
CO-4	Compare the performance of multiple methods and models, recognize the connections between how the data were collected and the scope of conclusions from the resulting analysis, and articulate the limitations and abuses of formal inference and modeling.	Analyze
CO-5	Evaluate strategies for constructing models and can use different measures of model fit and performance to assess models.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Data Science- Characteristics & Evolution of data, Data Science Process, Types & Levels of data, Datafication, Steps of Data Science, Central Tendency, Measure of Dispersion, Data Munging, Feature Engineering	**4**
2	Statistical Methods in Data Science- Data Distribution (Bernoulli, Uniform, Binomial, Normal, Poisson, Exponential), Mathematical Statistics, Inferential Statistics, Descriptive Statistics, Random Variable, Probabilistic Statistics, Sampling of data, Correlation Analysis	**5**
3	Technologies & Tools in Database Analytics- SQL Essentials for data science, String Pattern, Ranges, Sorting & Grouping Result Set, working with multiple tables, accessing database using R/Python,	**6**

	Database Text Analysis, User defined Functions & Aggregates, MADlib, Tools & Techniques for unstructured data.	
4	Computing techniques for Data Science- Regression, Mapping Problem to Machine Learning Task, Memorization Method, Generalized Additive Models, Time-Series Model, Predictive Modeling, Ensemble Techniques, Outlier Detection, Recommendation Engine	**9**
5	Statistical Methods for Evaluation- Hypothesis Testing, Difference of Means, Significance Level and P-Value, Test Statistics (Z-test, ANOVA, T-Test, Redundancy Test), Bias Variance Trade off, Cross Validation	**2**
6	Exploratory Data Analysis & Data Science Process- Visualization before analysis, Dirty Data, Visualizing single and multiple variables, summary statistics of EDA, Data Exploration versus Presentation, Real time case study, Tools & Techniques	**2**
Total lectures		**28**

Suggested Text Book(s):

1. Haider, M. (2015). Getting Started with Data Science: Making Sense of Data with Analytics. IBM Press.
2. Dietrich, D. (Ed.). (2015). Data science & big data analytics: discovering, analyzing, visualizing and presenting data. Wiley.
3. Trevor, H., Robert, T., & JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction.

Suggested Reference Book(s):

1. Grus, J. (2015). Data science from scratch: First principles with Python. " O'Reilly Media, Inc.".
2. Taylor, J. K., &Cihon, C. (2004). Statistical techniques for data analysis. Chapman and Hall/CRC.
3. Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
4. Zumel, N., & Mount, J. (2014). Practical data science with R. Manning Publications Co..
5. Saltz, J. S., & Stanton, J. M. (2017). An introduction to data science. SAGE Publications.

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	3	2	1	1	1	3	2	1	**1.8**
CO-2	3	3	3	3	3	1	1	2	2	2	1	**2.1**
CO-3	2	2	2	2	1	1	2	2	2	2	2	**1.8**
CO-4	3	3	3	3	2	1	1	3	2	3	2	**2.3**
CO-5	3	2	3	2	1	1	2	1	1	3	1	**1.8**
Average	**2.6**	**2.4**	**2.6**	**2.6**	**1.8**	**1.0**	**1.6**	**1.8**	**2.0**	**2.4**	**1.6**	

Computing for Data Science Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Python Programming

Course Objectives:

1. To implement data science workflows using Python/R.
2. To perform statistical analysis on real-world datasets.
3. To work with structured and unstructured data.
4. To apply machine learning models and evaluate them.
5. To visualize and interpret data effectively.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Apply data preprocessing and feature engineering techniques	Apply
CO2	Perform statistical analysis using programming tools.	Apply
CO3	Use SQL and database tools for data analytics	Analyze
CO4	Implement machine learning models	Apply
CO5	Evaluate models using statistical metrics and validation methods	Evaluate

List of Experiments

S. No	Description	Hours
1	Get acquainted with data science tools and perform statistical analysis	**2**
2	Hypothesis tests (e.g., t-tests, Z-tests) on sample datasets to compare population means	**2**
3	Simulate and apply the central limit theorem (CLT) to demonstrate how sample distributions converge to a normal distribution	**2**
4	Perform data wrangling and ETL processes on a dataset, followed by exploratory data analysis (EDA)	**2**
5	Utilize tools to create effective data visualizations (e.g., line charts, bar charts, heat maps, box plots) to derive key insights from the dataset	**4**
6	Implement feature extraction and selection techniques, including experimenting with encoding methods like one-hot encoding and creating new features based on domain expertise	**4**
7	Develop a simple linear regression model, extend it to multiple linear	**4**

	regression with several variables, and visualize both the regression line and residual plots	
8	Develop a simple linear regression model, extend it to multiple linear regression with several variables, and visualize both the regression line and residual plots	**4**
9	Apply K-means clustering and assess cluster quality using evaluation metrics like the silhouette score	**4**
Total Lab hours		**28**

Suggested Text Book(s):

1. Haider, M. (2015). Getting Started with Data Science: Making Sense of Data with Analytics. IBM Press.
2. Dietrich, D. (Ed.). (2015). Data science & big data analytics: discovering, analyzing, visualizing and presenting data. Wiley.
3. Trevor, H., Robert, T., & JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction.

Suggested Reference Book(s):

1. Grus, J. (2015). Data science from scratch: First principles with Python. " O'Reilly Media, Inc.".
2. Taylor, J. K., &Cihon, C. (2004). Statistical techniques for data analysis. Chapman and Hall/CRC.
3. Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
4. Zumel, N., & Mount, J. (2014). Practical data science with R. Manning Publications Co..
5. Saltz, J. S., & Stanton, J. M. (2017). An introduction to data science. SAGE Publications.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	3	2	1	1	1	3	2	1	**1.8**
CO-2	3	3	3	3	3	1	1	2	2	2	1	**2.1**
CO-3	2	2	2	2	1	1	2	2	2	2	2	**1.8**
CO-4	3	3	3	3	2	1	1	3	2	3	2	**2.3**
CO-5	3	2	3	2	1	1	2	1	1	3	1	**1.8**
Average	**2.6**	**2.4**	**2.6**	**2.6**	**1.8**	**1.0**	**1.6**	**1.8**	**2.0**	**2.4**	**1.6**	

Sensor Technology and Android Programming

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basic Programming, Mobile App Development (preferred)

Course Objectives:

1. To understand different types of sensors and their working principles.
2. To analyze sensor data and integrate it into mobile applications.
3. To develop context-aware Android applications using sensor APIs.
4. To explore external sensing technologies like NFC, RFID, and camera-based systems.
5. To design intelligent mobile services using sensor data.

Course Outcomes:

S. No.	Course Outcomes	Level
CO-1	Explain different types of sensors and smart sensor systems	Understand
CO-2	Analyze sensor data and the working of sensor-based systems	Analyze
CO-3	Apply Android Sensor APIs to develop context-aware applications	Apply
CO-4	Evaluate external sensing technologies such as NFC, RFID, and camera-based systems	Evaluate
CO-5	Design intelligent mobile applications using multiple sensors	Create

Course Contents:

Unit	Contents	Lectures
1	Fundamentals of Sensors and Smart Systems: Sensing concepts, sensor characteristics, sensing modalities; Mechanical, MEMS, Optical, Semiconductor, Electrochemical, Biosensors (overview only); Smart sensors, sensor systems, microcontrollers (Arduino overview), sensor data acquisition basics	6
2	Android for Sensor-Based Applications (Advanced): Quick recap of Android components (Activity, Manifest – brief), Android architecture for sensor apps, permissions, background services, introduction to sensor framework (avoid UI basics), handling real-time data	4

3	Android Sensor API and Physical Sensing: Types of sensors (accelerometer, gyroscope, proximity, light, pressure), SensorManager, sensor events, motion detection, orientation sensing, environmental sensing, data filtering basics	6
4	External and Augmented Sensing Systems: RFID, NFC basics, NFC-based applications (payments, tagging), camera as sensor (barcode scanning, QR code), introduction to image processing concepts, Android Open Accessory (AOA) overview and limitations	6
5	Sensor-Based Intelligent Services: Context-aware applications, motion detection apps, air quality monitoring (concept), smart brightness control, activity recognition, location + sensor fusion basics, energy-efficient sensing	6
	Total Lectures	**28**

Suggested Text Books:

1. Krzysztof Iniewski, *Smart Sensors for Industrial Applications*
2. Bill Phillips et al., *Android Programming: The Big Nerd Ranch Guide*

Suggested Reference Books:

1. Mike Banzi, *Getting Started with Arduino*
2. Android Developer Documentation

Other Resources:

1. YouTube

a) Android Sensors Tutorial (Accelerometer, Gyroscope, SensorManager)
b) Android App Development Full Course

2. Coursera

a) Android App Development Specialization (Vanderbilt University)
b) Introduction to the Internet of Things and Embedded Systems (University of California, Irvine)

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered up to T-1
2	T-2	25	1.5 Hours	Syllabus covered up to T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	2	1	1	1	1	2	1.4
CO-2	3	3	2	3	2	2	1	1	1	1	2	1.8
CO-3	3	3	3	2	3	2	1	1	2	1	2	2
CO-4	2	3	2	2	3	2	1	1	2	2	2	2
CO-5	3	2	3	2	3	3	1	2	2	2	3	2.3
Average	2.8	2.6	2.2	2	2.6	2.2	1	1.2	1.6	1.4	2.2	

Sensor Technology and Android Programming Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

L-T-P: 0-0-2

Pre-requisite: Basic Programming, Mobile App Development (preferred)

Course Objectives:

1. To provide hands-on experience with Android sensor APIs.
2. To develop applications using mobile and external sensors.
3. To analyze real-time sensor data.
4. To build context-aware and intelligent applications.
5. To integrate sensing technologies into mobile apps.

Course Outcomes:

S. No.	Course Outcomes	Level
CO-1	Implement applications using Android sensor APIs	Apply
CO-2	Analyze and process sensor data from mobile devices	Analyze
CO-3	Develop applications using motion and environmental sensors	Apply
CO-4	Implement applications using NFC, camera, and external sensing	Apply
CO-5	Design a sensor-based intelligent mobile application	Create

List of Experiments:

S. No	Description	Hours
1	Study and implement listing of available sensors in an Android device using SensorManager	2
2	Develop an application to read and display accelerometer sensor data in real time	2
3	Implement an application for motion detection using accelerometer sensor data	2
4	Develop an application using gyroscope sensor for device orientation detection	2
5	Implement an application using proximity and light sensors for smart control (auto brightness concept)	2
6	Develop an application using pressure/barometer sensor for	2

	environmental sensing (or simulation if unavailable)	
7	Implement sensor data logging and visualization using graphical representation	2
8	Develop an application using GPS and sensor fusion for location and motion tracking	2
9	Implement a step counter / activity recognition application using motion sensors	2
10	Develop an NFC-based application for reading NFC tags (or simulation if hardware not available)	2
11	Implement a camera-based application for QR code/barcode scanning	2
12	Implement integration of external sensors (Arduino + temperature sensor) with Android (demo/simulation)	2
13	Implement a context-aware service	2
14	Design a sensor-based intelligent application	2
Total Lab Hours		**28**

Suggested Text Books:

1. Krzysztof Iniewski, *Smart Sensors for Industrial Applications*
2. Bill Phillips et al., *Android Programming: The Big Nerd Ranch Guide*

Suggested Reference Books:

1. Mike Banzi, *Getting Started with Arduino*
2. Android Developer Documentation

Other Resources:

1. NFC Forum – **https://nfc-forum.org**
2. ML Kit – **https://developers.google.com/ml-kit**
3. Arduino – **https://www.arduino.cc/en/Tutorial/HomePage**
4. MIT App Inventor – https://appinventor.mit.edu

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects, etc.)	60

	Total	100

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	2	1	1	1	1	2	1.7
CO-2	3	3	2	3	3	2	1	1	1	1	2	1.9
CO-3	3	2	3	2	3	2	1	2	2	1	2	2
CO-4	3	2	3	2	3	2	1	2	2	1	2	2
CO-5	3	3	3	3	3	2	1	2	2	2	3	2.4
Average	3	2.4	2.6	2.4	3	2	1	1.6	1.6	1.2	2.2	

Concept of Graph Theory

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Discrete Mathematics

Course Objectives:

1. To understand fundamental concepts of graph theory including graphs, subgraphs, and representations.
2. To apply traversal and connectivity techniques in solving graph-related problems. To apply shortest path and spanning tree algorithms
3. To analyze properties of trees, cycles, and planar graphs.
4. Analyze Graph embeddings.
5. To design solutions using graph theory concepts in real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Describe basic concepts of graphs, types of graphs, and their representations.	Understand
CO-2	Apply graph traversal techniques and connectivity properties. Apply graph algorithms such as BFS, DFS, shortest path, and spanning tree algorithms.	Apply
CO-3	Analyze matching, coloring, and planar graphs for structural properties.	Apply
CO-4	Analyze Graph embeddings	Analyze
CO-5	Design solutions for real-world problems using graph theory concepts.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Graphs: Fundamental Concepts: Paths, cycles, and trails , Eulerian Graphs, Hamiltonian Graphs, Vertex degrees and counting , Directed graphs	**3**
2	Trees and Spanning Trees: Basic properties of trees, Spanning trees and enumeration, Optimization using trees	**4**
3	Matchings and Factors: Matchings and vertex covers, Maximum matching in Bipartite Graphs, Matchings in general graphs	**4**
4	Graph Traversal, Shortest Path Algorithms	**4**
5	Graph Connectivity and Cycles: Connectivity and Paths , Cuts and connectivity , k-connected graphs, Network flow problems, Ford-Fulkerson algorithm	**5**
6	Planar Graphs and Coloring: Coloring of Graphs, Vertex coloring and bounds, Structure of k-chromatic graphs, Enumeration aspects, Graph	**8**

	embeddings, Euler's formula, Characterization of planar graphs, Planarity parameters Edges and Cycles: Line graphs and edge coloring, Hamiltonian cycles Relations between planarity, coloring, cycles	
Total lectures		**28**

Suggested Text Book(s):

4. Introduction to Graph Theory: D.B. West (2001) Prentice Hall.
5. Introduction to Algorithms by Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein, 3rd ed. MIT Press, 2009. ISBN: 9780262033848.
6. Graph Theory with applications to engineering and computer science Narsingh Deo – Graph Theory

Suggested Reference Book(s):

1. R.Diestel: Graph Theory,Springer(low price edition) 2000.

Other useful resource(s):

6. Link to NPTEL course contents: NPTEL Link: Graph Theory, By Prof. SoumenMaity | IISER Pune
2. Link to topics related to course:
   a. Stanford CS224W: Machine Learning with Graphs

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	1	1	1	1.5
CO-2	3	3	2	2	2	1	1	1	2	1	1	1.9
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	2	3	2	3	2	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	1	3	2	2	2.1
Average	2.8	2.6	2.2	2.0	2.4	1.0	1.0	1.0	2.0	1.6	1.6	

Concept of Graph Theory Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Programming in C++, Data Structures

Course Objectives:

1. To implement basic graph representations and operations using programming.
2. To apply graph traversal and shortest path algorithms programmatically.
3. To analyze structural properties like trees, matchings, and connectivity using code.
4. To simulate planar graphs, coloring, and embeddings computationally.
5. To develop problem-solving skills using graph-based real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	Implement graph representations and basic operations.	Apply
CO2	Apply traversal and shortest path algorithms.	Apply
CO3	Analyze trees, connectivity, and matchings	Analyze
CO4	Implement graph coloring and planarity checks.	Analyze
CO5	Design real-world applications using graph models	Create

List of Experiments

S. No	Description	Hours
1	To implement graph using adjacency matrix and adjacency list. Degree of vertices and checking graph properties (directed/undirected) Implementation of Eulerian path and circuit detection Check Hamiltonian path (backtracking approach)	**2**
2	Implementation of tree properties and traversal, Minimum Spanning Tree using Kruskal's Algorithm, Minimum Spanning Tree using Prim's Algorithm	**2**
3	Implement maximum matching in bipartite graph (Hungarian / BFS based), Implement Vertex cover approximation	**2**
4	Implement Breadth First Search (BFS), Depth First Search (DFS), Dijkstra's shortest path algorithm	**2**
5	Project	**2**
6	Implement check connectivity and articulation points, Implement Ford-Fulkerson algorithm for maximum flow	**4**

7	To implement graph coloring using backtracking. To simulate real-world map coloring.	**4**
8	To implement Planar Graph Visualization, Create dual graph from planar graph	**4**
9	To implement graph embedding for machine learning (node embeddings)	**6**
Total Lab hours		**28**

Suggested/Resources:

1. West, D.B., Introduction to Graph Theory, Prentice Hall, 2001.
2. Diestel, R., Graph Theory, Springer, 2000.
3. Cormen, T. et al., Introduction to Algorithms, MIT Press, 2009.
4. Deo, N., Graph Theory with Applications, PHI.

Other useful resource(s):

11. Link to NPTEL Graph Theory Course: Covers: Planarity, embeddings, connectivity
12. Stanford CS224W: Covers: Node embeddings, graph ML

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	3	1	1	1	1	1	1	1.6
CO-2	3	3	2	2	3	1	1	1	2	1	1	1.9
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	2	3	2	3	3	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	1	3	2	2	2.1
Average	2.8	2.6	2.2	2	3	1	1	1	2	1.6	1.6	

Computational Data Analysis

COURSE CODE: XXXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: Elective

L-T-P: 2-0-0

Pre-requisite: Basics of probability and statistics

Course Objectives:

1. To provide students with foundational knowledge of topics such as probability and statistics and algorithms to solve data analysis problems arising in practical applications,
2. To enable students to effectively apply computational methods to solve exemplar data analysis problems arising in relevant applications.
3. Have a good understanding of the fundamental issues and challenges of machine learning.
4. Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
5. Be able to design and implement various machine learning algorithms in a range of real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn and apply supervised learning techniques including regression and ensemble learning for predictive modeling.	Apply
CO-2	To understand and implement probabilistic and generative models for data modeling and classification tasks.	Apply
CO-3	To learn and use clustering and unsupervised learning techniques for discovering patterns in unlabeled data.	Analyze
CO-4	To learn and use various techniques for feature selection and dimensionality reduction to improve data representation.	Analyze
CO-5	To learn about various techniques for model optimization, regularization, and performance enhancement of machine learning algorithms.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	Discriminative Algorithms: Concept of supervised learning, linear regression, parameter estimation using maximum likelihood estimation, normal equation method, gradient descent, stochastic gradient descent, stochastic variance reduced gradient. Linear classification, logistic regression, multiclass	6

	classification using one versus rest and one versus one approach, multinomial regression, k nearest neighbor. Support vector machines, geometric margins, kernel methods.	
2	Generative Algorithms: Multivariate normal distribution, Naive Bayes classifier, Laplacian smoothing, linear discriminant analysis, multi class Fisher discriminant analysis, Bayesian regression, Bayesian logistic regression.	**5**
3	Unsupervised Learning: Partition based clustering: k means clustering, Hierarchical clustering: agglomerative clustering and divisive clustering, Density based clustering: density based spatial clustering of applications with noise, Model based clustering: expectation maximization algorithm using Gaussian mixture models	**5**
4	Regularization, Model Selection and Optimization: Concept of overfitting and bias variance tradeoff, ridge regression, least absolute shrinkage and selection operator regression, elastic net regression, forward selection, backward elimination, cross validation, Bayesian optimization. Proximal gradient methods, Proximal stochastic variance reduced gradient.	**5**
5	Feature Engineering and Dimensionality Reduction: Feature selection using information gain, correlation coefficient, chi square test, dimensionality reduction using principal component analysis.	**4**
6	Ensemble Learning: Bagging: Random Forest, Boosting: Adaptive Boosting, Gradient Tree Boosting, Stacking.	**3**
	Total lectures	**28**

Suggested Text Book(s):

1. Tom Mitchell, "Machine Learning", McGraw Hill, 1997, ISBN 0070428077
2. Introduction to Machine Learning Edition 2, by EthemAlpaydin.
3. T. Hastie, R. Tibshirani, & J. H. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer Verlag, 2001.

Suggested Reference Book(s):

1. The Elements of Statistical Learning, 2nd Edition, Hastie, Tibshirani and Friedman.
2. Foundations of Machine Learning, Mohri, Rostamizadeh and Talwalker
3. Richard o. Duda, Peter E. Hart, and David G. Stork, "Pattern Classification", John Wiley Asia, 2006

Other useful resource(s):

1.Link to NPTEL course contents:https://nptel.ac.in/courses/106106139/
2. https://nptel.ac.in/courses/110106064/

EvaluationScheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.73**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	2	1	1	1	2	3	2	**2.09**
CO-5	1	2	2	1	3	2	1	1	1	1	1	**1.45**
Average	**1.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.2**	**1**	**1**	**1.8**	**2**	**1.4**	

Computational Data Analysis Lab

COURSE CODE: XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Python Programming Language

Course Objectives:

1. To develop students' skill in software development techniques using one or more high level programming languages relevant to data analytics.
2. To enable students to effectively apply computational methods to solve exemplar data analysis problems arising in relevant applications.
3. To learn the implementation of classification techniques for any dataset.
4. To conduct experiments for clustering techniques for any dataset.
5. To discuss different classification and clustering algorithms based on the analysis of results obtained from experimental evaluation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO1	To apply the basic concepts and statistical techniques used in machine learning using Python.	Apply
CO2	To apply classification and clustering algorithms for data analysis using Python.	Apply
CO3	To apply feature selection methods for improving model performance using Python.	Apply
CO4	To apply different machine learning algorithms and compare their performance using suitable evaluation metrics.	Apply, Analyze

List of Experiments

S. No	Description	Hours
1	Revision of Python programming using NumPy and Pandas for data handling and manipulation.	**2**
2	Data visualization using Matplotlib and Seaborn libraries.	**2**
3	Implementation of linear regression using normal equation and gradient descent.	**2**

4	• Implementation of logistic regression for classification tasks. • Implementation of k nearest neighbors algorithm.	**2**
5	Implementation of support vector machines with linear and kernel methods.	**2**
6	Implementation of naive Bayes classifier with Laplacian smoothing.	**2**
7	Implementation of k means and hierarchical clustering algorithm.	**2**
8	• Implementation of density based spatial clustering of applications with noise. • Implementation of expectation maximization algorithm using Gaussian mixture models	**2**
9	Implementation of regularization techniques including ridge regression and least absolute shrinkage and selection operator regression.	**2**
10	Implementation of feature selection techniques and principal component analysis.	**4**
11	Implementation of ensemble learning methods including random forest, adaptive boosting, and gradient boosting.	**2**
12	Mini project	**4**
Total Lab hours		**28**

Suggested/Resources:

1. Tom Mitchell, “Machine Learning”, McGraw Hill, 1997, ISBN 0070428077
2. Sebastian Raschka, “Python Machine Learning”, Packt Publishing Ltd.
3. Andreas C. Müller, Sarah Guido, “Introduction to Machine Learning with Python”, O'Reilly Media, Inc.
4. SunilaGollapudi, “Practical Machine Learning”, Packt Publishing Ltd
5. Wes McKinney, “Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython”, O'Reilly Media, Inc.
6. Willi Richert, “Building Machine Learning Systems with Python”, Packt Publishing Ltd.
7. Link to topics related to course:

xiii. https://www.python-course.eu/machine_learning.php

xiv. https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/

xv. https://www.datacamp.com/

Evaluation Scheme:

S. No	**Exam**	**Marks**
1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records,	60

	experiment execution, viva voce, quizzes, mini-projects etc.)	
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	**1.73**
CO-2	3	3	3	3	3	2	1	2	2	1	2	**2.27**
CO-3	3	3	2	3	3	2	1	2	2	1	2	**2.18**
CO-4	3	3	3	3	3	2	1	2	2	1	2	**2.27**
Average	**3**	**2.75**	**2.5**	**2.75**	**3**	**1.75**	**1**	**1.75**	**1.75**	**1**	**2**	

Embedded Software Development

COURSE CODE: XXXXXXX

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: C Programming

Course Objectives:

1. Introduce embedded software architecture, development lifecycle, Embedded C programming, low-level firmware design, and hardware-interfacing modules.
2. Enable students to understand real-time constraints, RTOS-based design, along with debugging, testing, and deployment tools for embedded systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the structure of embedded software systems, including architectural layers, system constraints, and the embedded software development lifecycle.	Understand
CO-2	Apply RTOS concepts to design multitasking embedded systems using scheduling algorithms and inter-process communication mechanisms.	Apply
CO-3	Utilize debugging, testing, and development tools to analyze, verify, and optimize embedded software systems.	Analyze
CO-4	Develop embedded firmware using C by applying register-level programming, memory-mapped I/O, and efficient bit manipulation techniques.	Create
CO-5	Design and implement interrupt-driven programs to handle asynchronous events while ensuring real-time responsiveness.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Embedded Software: Embedded systems overview (hardware–software co-design); Role of software in embedded systems; Embedded vs general-purpose software; Embedded software stack: Bootloader, Device drivers, Middleware, Application layer. Embedded system constraints: real-time, memory, power	4
2	Embedded C Programming & Firmware Basics: Review of C for embedded systems, Embedded C vs standard C, Data types, memory models, pointers. Bit manipulation and register-level programming, Memory-mapped I/O, Firmware structure and coding standards, Interrupt Service Routines (ISR)	6

3	Microcontroller Software Interface: Overview of microcontroller peripherals, GPIO programming, Timers and counters, ADC/DAC interfacing, Serial communication: UART, SPI, I2C. Device driver basics	5
4	Real-Time Embedded Software & RTOS: Real-time systems: hard vs soft real-time, Task, thread, process concepts. Scheduling algorithms (RMS, EDF basics). RTOS architecture, Inter-process communication (IPC): Semaphores, Message queues, Mutex, Context switching and latency	6
5	Embedded Software Design & Development Process: Embedded software development life cycle (ESDLC), Requirement analysis and specification, State machine and event-driven programming, Hardware–software co-design, Code optimization techniques, Power-aware software design	4
6	Debugging, Testing & Tools: Debugging techniques - JTAG, ICE, simulators; Unit testing and integration testing, Static and dynamic analysis, IDEs and toolchains (Keil, MPLAB, GCC, etc.), Version control basics (Git for embedded projects)	3
Total lectures		28

Suggested Text Book(s)

1. David E. Simon, "Embedded Software Primer," Pearson Education, 2002.
2. Michael Barr, "Programming Embedded Systems: in C and C++," Shroff Publishing & Distributors, 2003.
3. Raj Kamal, "Embedded Systems," McGraw Hill, 2020.

Suggested Reference Book(s)

1. Qing Li, Caroline Yao "Real-Time Concepts for Embedded Systems," Taylor&Francis, 2003.
2. Jonathan Valvano, "Embedded Systems: Introduction to ARM Cortex-M Microcontrollers," Createspace Independent Pub, 2012.
3. Marilyn Wolf, "Computers as Components: Principles of Embedded Computing System Design," A Trueman Publication, 2022.

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc20_ee98/preview
2. Link to NPTEL course contents:https://nptel.ac.in/courses/106105159
3. Link to NPTEL course contents: https://nptel.ac.in/courses/106105193
4. Link to coursera course contents: https://www.coursera.org/learn/introduction-embedded-systems

Evaluation Scheme:

S. No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	1	1	1	1	1	1	**1.3**
CO-2	3	3	3	2	2	1	1	2	1	1	2	**1.9**
CO-3	2	3	2	3	3	1	1	1	1	1	2	**1.8**
CO-4	3	3	2	1	3	1	1	1	1	1	1	**1.6**
CO-5	3	3	2	2	2	1	1	1	1	1	1	**1.6**
Average	**2.8**	**2.8**	**2**	**1.8**	**2.2**	**1**	**1**	**1.2**	**1**	**1**	**1.4**	

Embedded Software Development Lab

COURSE CODE: XXXXXXX
COURSE CREDITS: 1
CORE/ELECTIVE: ELECTIVE
L-T-P: 0-0-2

Pre-requisite: C Programming

Course Objectives:

1. To provide hands-on experience in embedded C programming and microcontroller-based system design, including the development, interfacing, and integration of sensors, and various peripherals.
2. To develop proficiency in real-time embedded systems concepts, covering interrupts, timers, communication protocols, along with embedded software debugging.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Write, compile, and deploy efficient embedded C programs tailored for microcontroller architectures to control hardware operations effectively.	Apply
CO-2	Interface various digital and analog sensors, actuators, and peripherals with microcontrollers to enable seamless data acquisition and control in embedded applications.	Apply
CO-3	Implement timers, interrupts, and real-time control techniques to manage time-critical tasks and ensure deterministic behavior in embedded systems.	Apply
CO-4	Design, simulate, test, and debug embedded applications using development tools, IDEs, and debugging techniques for reliable system performance.	Apply
CO-5	Implement and utilize standard communication protocols such as UART, SPI, I2C, and others to enable efficient data exchange between microcontrollers and peripheral devices.	Apply

List of Experiments:

S. No.	**Details**	**Lab Hours**
1	Introduction to Embedded IDE and Toolchain: To install and get familiar with the embedded development environment, including the IDE, compiler, debugger, and toolchain. Execute a basic "Hello World" and LED blinking program to understand the complete build and download process.	2
2	GPIO Programming: To program the General Purpose Input/Output (GPIO) pins of a microcontroller to interface and control an LED and a push button, enabling basic digital input and output operations.	2
3	Software Delay Generation: Generate precise time delays using software loops in embedded C and observe the effect of different loop counts on delay duration, understanding the limitations of software-based timing.	2
4	Timer Programming: To configure and use hardware timers of the microcontroller to generate accurate delays and time intervals, demonstrating the superiority of hardware timing over software loops.	2

5	External Interrupt Handling: To implement external interrupt handling to achieve event-driven control, such as responding immediately to a push button press without continuous polling.	2
6	Timer Interrupt-Based Task: To configure timer interrupts to execute periodic tasks, enabling multitasking-like behavior and precise timing control in embedded applications..	2
7	LCD Interfacing: To interface a character LCD module with the microcontroller and write programs to display text, numbers, and custom characters on the LCD screen.	2
8	ADC Interfacing: To interface an Analog-to-Digital Converter (ADC) to read values from analog sensors (such as potentiometer or temperature sensor) and display the converted digital data.	2
9	PWM Generation: To generate Pulse Width Modulation (PWM) signals to control the brightness of an LED, demonstrating the use of PWM for variable power control and motor speed regulation.	2
10	UART Communication: To implement UART serial communication to transmit and receive data between the microcontroller and a PC or another device, enabling real-time data exchange.	2
11	I2C Communication: To interface peripheral devices using the I2C protocol and perform master-slave communication for reading/writing data from sensors or memory devices..	2
12	Mini Project – Temperature Monitoring System. To design and implement a complete embedded application for a temperature monitoring system using sensors, display, and appropriate control logic, integrating concepts learned throughout the lab.	2
	Total LAB Hours	**24**

Suggested Text Book(s)

1. David E. Simon, "Embedded Software Primer," Pearson Education, 2002.
2. Michael Barr, "Programming Embedded Systems: in C and C++," Shroff Publishing & Distributors, 2003.
3. James M. Fiore, "Embedded Controllers Using C and Arduino," 2016.

Suggested Reference Book(s)

1. Michael Barr, Anthony Massa, "Programming Embedded Systems," O′Reilly, 2006.
2. Jonathan Valvano, "Embedded Systems: Introduction to ARM Cortex-M Microcontrollers," Createspace Independent Pub, 2012.

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc20_ee98/preview
2. Link to NPTEL course contents:https://nptel.ac.in/courses/106105159
3. Link to NPTEL course contents: https://nptel.ac.in/courses/106105193
4. Link to coursera course contents: https://www.coursera.org/learn/introduction-embedded-systems

Evaluation Scheme:

S. No	Exam	Marks

1	Mid Sem. Evaluation (P-1)	20
2	End Sem. Evaluation (P-2)	20
3.	Internal Assessment (P-3) (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	60
	Total	**100**

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	3	1	1	1	1	1	1	**1.5**
CO-2	3	2	2	1	3	1	1	1	1	1	1	**1.5**
CO-3	3	3	2	2	3	1	1	1	1	1	1	**1.7**
CO-4	3	3	3	2	3	1	1	1	2	2	1	**2.0**
CO-5	3	2	2	1	3	1	1	1	1	2	1	**1.6**
Average	**3**	**2.4**	**2**	**1.4**	**3**	**1**	**1**	**1**	**1.2**	**1.4**	**1**	

Introduction to Deep Learning

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning, Linear Regression, Probability Theory

Course Objectives:

1. To build a foundational understanding of neural network architectures, activation functions, and feedforward computation.
2. To develop proficiency in training deep networks using backpropagation, gradient-based optimizers, and regularization strategies.
3. To study convolutional and recurrent architectures for image and sequence modelling tasks, along with pre-trained model ecosystems.
4. To introduce the attention mechanism and Transformer-based models including BERT and GPT for language understanding and generation.
5. To expose learners to advanced generative models, post-Transformer architectures, and the latest trends in Generative AI.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of artificial neural networks	Understand
CO-2	Apply gradient descent and backpropagation while analyzing convergence and overfitting.	Apply, Analyze
CO-3	Design CNNs with transfer learning for vision tasks.	Apply, Evaluate
CO-4	Develop sequence models using RNNs, LSTM, GRU, and transformers.	Apply, Analyze
CO-5	Evaluate advanced models such as GANs, autoencoders, diffusion models, and modern large-scale architectures.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	Introduction to biological and artificial neurons, linear perceptron model and its representation as a neuron, decision boundaries and limitations of linear models including XOR problem, feedforward neural networks, activation functions such as Sigmoid, Tanh and ReLU, and Softmax output layer. Measurement of errors using loss functions including Mean Squared Error and Cross-Entropy, gradient descent optimization methods including batch, stochastic and mini-batch approaches, delta rule and learning rate, backpropagation algorithm, bias-variance tradeoff, overfitting and its prevention using regularization techniques such as L1, L2, ElasticNet dropout and early stopping, and advanced optimization algorithms including Adagrad, RMSProp, Adadelta and	**5**

	Adam.	
2	Motivation from human visual perception and limitations of traditional feature engineering approaches, scaling issues in fully connected neural networks, convolution operation including filters, stride and padding, feature maps and hierarchical feature learning, pooling operations, convolutional neural network architectures (CNNs) and design principles, applications in image classification and recognition tasks.	**7**
3	Self-supervised pretraining approaches, overview of classical deep learning architectures including AlexNet, VGG, GoogleNet (Inception), ResNet and DenseNet, transfer learning techniques including feature extraction and fine-tuning, and introduction to few-shot learning for limited data scenarios.	**5**
4	Introduction to sequential data and embedding techniques, basic recurrent neural network architecture (RNN) and hidden state representation, challenges of vanishing and exploding gradients, advanced recurrent architectures including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), bidirectional and deep recurrent neural networks	**7**
5	Limitations of recurrent models for long-range dependencies, attention mechanisms, self-attention and multi-head attention, positional encoding techniques, transformer architecture including encoder-decoder structure, and pretrained transformer models such as BERT.	**5**
6	Generative Adversarial Networks including generator and discriminator architecture, variants of GANs, autoencoders including sparse and denoising autoencoders.	**6**
7	Scaling laws and foundation models, large language models and their architectural overview, efficient transformer variants including sparse and linear attention mechanisms, retrieval-augmented generation techniques.	**7**
Total lectures		**42**

Suggested Text Book(s):

1. Ian Goodfellow, YoshuaBengio, and Aaron Courville, “Deep Learning”, The MIT Press, Cambridge, 2016.
2. Charu C. Aggarwal, “Neural Networks and Deep Learning: A Textbook”, Springer, 2018.

Suggested Reference Book(s):

1. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", 2nd Edition, OReillyMedia.
2. Andrew Ng, “Machine Learning Yearning”, available online published by deeplearning.ai

Other useful resource(s):

Link to NPTEL course contents:

1. https://www.youtube.com/watch?v=aPfkYu_qiF4&list=PLyqSpQzTE6M9gCgajvQbc68Hk_JKGBAYT
2. https://www.youtube.com/watch?v=oJlb4jBbKWw&list=PLyqSpQzTE6M-_1jAqrFCsgCcuTYm_2urp
3. https://www.youtube.com/watch?v=qrsNX1Rwle0&list=PLZ2ps__7DhBbaMNZoyW2Hizl8DG6ikkjo

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	1	1	1	1	2	2	1	**1.5**
CO-2	3	3	1	2	2	1	1	1	2	2	1	**1.7**
CO-3	2	2	3	2	3	1	1	1	2	2	1	**1.8**
CO-4	2	3	2	2	3	1	1	1	2	3	2	**2.0**
CO-5	2	2	1	2	2	2	1	1	1	1	3	**1.6**
Average	**2.4**	**2.4**	**1.6**	**1.8**	**2.2**	**1.2**	**1**	**1**	**1.8**	**2**	**1.6**	

Cryptocurrency Technologies

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: C++/Python, data structures,discrete mathematics

Course Objectives:

1. To understand fundamental cryptographic primitives used in cryptocurrencies.
2. To explain the working of blockchain and distributed consensus mechanisms.
3. To analyze the Bitcoin protocol and mining process.
4. To explore security, privacy, and anonymity issues in cryptocurrencies.
5. To study real-world applications such as smart contracts and decentralized finance.
6. To develop the ability to design and evaluate secure cryptocurrency systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain various cryptographic techniques like hashing and digital signatures.	Understand
CO-2	Explain the structure and functioning of blockchains and Merkle trees.	Understand
CO-3	Analyze and implement Bitcoin-like protocols.	Apply
CO-4	Evaluate security risks, attacks, and privacy mechanisms in cryptocurrencies.	Analyze
CO-5	Design basic decentralized applications and smart systems. Critically assess future trends and innovations in block-chain technology.	Create

Course Contents:

Unit	Contents	Lectures required
1	Evolution of Digital currency; limitations of traditional banking , Introduction to Bitcoin; overview of decentralized systems; role of trust and intermediaries, Centralized vs decentralized vs distributed systems; advantages and challenges , Applications of block-chain: finance, supply chain, healthcare, governance; legal and regulatory overview.	**4**
2	Fundamentals of cryptography: confidentiality, integrity, authentication, non-repudiation, symmetric vs asymmetric cryptography; RSA and key exchange basics, Public key infrastructure (PKI) and key management, Digital signatures: algorithms, verification, security properties, Cryptographic hash functions: SHA family, properties (collision resistance, avalanche effect), Hash pointers and tamper-evident structures, Merkle trees: construction, proof of inclusion, efficiency, Applications of Merkle trees in blockchain verification and scalability	**8**
3	Cryptographic Data Structures & Blockchain:Blockchain fundamentals: blocks, headers, transactions, timestamps, Append-only ledger and	**7**

	immutability; trustless systems, Distributed ledger technology (DLT) and peer-to-peer networks, Transaction lifecycle: creation, propagation, validation ,Consensus basics: agreement in distributed systems; Byzantine fault tolerance, Forks (soft vs hard), longest chain rule, and network consistency.	
4	Bitcoin architecture and working: Keys and addresses as identities , Transactions and scripting basics, UTXO model and transaction lifecycle, Proof of Work (PoW) and mining process, Incentives and game theory in Bitcoin, Mining hardware: ASICs and mining pools , Alternative consensus mechanisms (PoS, Peercoin)	**5**
5	Bitcoin block structure: block header, Merkle root, nonce, Wallet technologies: hot wallets, cold wallets, hardware wallets, Key management: multi-signature, secret sharing, threshold cryptography , Financial transparency tools: proof of reserves, proof of liabilities	**5**
6	Security and Attacks: Threat model in cryptocurrencies; attack surfaces, Double spending attack and prevention mechanisms attack, selfish mining, eclipse attacks , Network security: Sybil attacks, DDoS, countermeasures and best practices.	**4**
7	Privacy and Anonymity : Privacy concepts: anonymity, pseudonymity, unlinkability, Transaction graph analysis and tracing techniques, De-anonymization attacks at network and application layer, Privacy-enhancing technologies: mixers, CoinJoin, blind signatures, zero-knowledge proofs	**6**
8	Advanced Applications & Future Trends : Smart contracts and smart property (e.g., Ethereum basics), Micropayments, payment channels (Lightning Network), auctions, voting systems, prediction markets, Scalability challenges, regulatory issues, future research directions and innovations	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder, Jeremy ClarkBitcoin and Cryptocurrency Technologies
2. Satoshi NakamotoBitcoin: A Peer-to-Peer Electronic Cash System
3. Michael NielsenHow the Bitcoin Protocol Actually Works

Suggested Reference Book(s):

1. Andreas M. Antonopoulos & Gavin WoodMastering Ethereum: Building Smart Contracts and DAppsImran BashirMastering Blockchain
2. Paul Vigna & Michael J. CaseyThe Age of Cryptocurrency
3. Narayanan Arvind et al. Additional lecture notes and Princeton course materials
4. Michael NielsenHow the Bitcoin Protocol Actually Works

Other useful resource(s):

1. Link to NPTEL course contents:https://www.youtube.com/watch?v=mzPoUjQC4WU&list=PLHRLZtgrF2jl8yqucJsMFqh5XpRLTgCI4
2. Link to NPTEL course contents:https://www.youtube.com/watch?v=GstOwbLyeYE&pp=ygUnQ3J5cHRvY3VycmVuY3kgVGVjaG5vbG9naWVzIGNvdXJzZSBucHRs

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Information Retrieval and Semantic Web

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Java/Python. Knowledge of data structures and algorithms.

Course Objectives:

1. To provide a comprehensive understanding of information retrieval principles and Semantic Web technologies.
2. To learn and analyze the information retrieval techniques.
3. To apply the information retrieval algorithms for real time applications.
4. To understand and evaluate the applications of information retrieval techniques

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Apply fundamental information retrieval models and techniques for efficient data retrieval.	Apply
CO-2	Apply ontology development methods and Semantic Web technologies for knowledge representation**.**	Understand
CO-3	Analyze query processing, clustering, and indexing techniques for large-scale data.	Apply
CO-4	Evaluate semantic web services and intelligent retrieval systems for real-world applications.	Analyze
CO-5	To understand and evaluate the applications of information retrieval techniques.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Information Retrieval (IR) and its applications: IR process and architecture, Classic IR models (Boolean, Vector Space, Probabilistic), Structured and unstructured text retrieval, Retrieval evaluation (precision, recall, F-measure), Evaluation metrics: precision, recall, F1-score, MAP, NDCG.	**5**
2	Query languages: keyword-based, structured queries (SQL-like, XML queries), Pattern matching and text search techniques, Query operations: intersection, union, ranking, filtering, Relevance feedback (explicit and implicit), Rocchio algorithm, Query expansion, synonym handling, multimedia query languages	**5**
3	Document preprocessing: tokenization, normalization, stop-word removal, Stemming and lemmatization techniques , Text compression: dictionary-based and statistical methods, Clustering techniques: hierarchical, k-means, evaluation of clusters, Inverted index: construction, storage, optimization, Ranking algorithms (TF-IDF, cosine	**6**

	similarity) and IR user interfaces	
4	Multimedia IR: concepts and challenges:Data models for multimedia (image, audio, video retrieval), Feature extraction: color, texture, shape descriptors, Web search engines: crawling, indexing, ranking (PageRank overview), Distributed IR: meta-search engines, parallel processing, scalability.	**5**
5	Evolution from Web 1.0 to Web 3.0 (Semantic Web): Limitations of syntactic web and need for semantics, Semantic Web architecture (layered model), Metadata and knowledge representation techniques, Ontologies: taxonomy, thesaurus, ontology classification	**5**
6	Resource Description Framework (RDF): triples, graphs, syntax, RDF Schema (RDFS): classes, properties, inheritance, Web Ontology Language (OWL): Lite, DL, Full, Ontology components: classes, properties, individuals, data types, Rule languages: RuleML, SWRL; inference and reasoning, Semantic reasoning: forward chaining, backward chaining, inference problems	**6**
7	Ontology development methodologies (Uschold& King, TOVE): Ontology editors: Protégé and tools for ontology creation, Ontology libraries and upper ontologies, Reasoners: Pellet, Hermit; consistency checking and classification, Semantic desktop, software agents, and intelligent information systems	**5**
8	Web services fundamentals: SOAP, REST, WSDL: Semantic Web services: OWL-S and service ontology, Service discovery, composition, and execution, Applications: geospatial systems, knowledge graphs, AI agents, Challenges: scalability, interoperability, big data integration, future trends	**5**
	Total lectures	**42**

Suggested Text Book(s):

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze
   Introduction to Information Retrieval
2. Grigoris Antoniou & Frank van Harmelen
   A Semantic Web Primer

Suggested Reference Book(s):

1. Ricardo Baeza-Yates & Berthier Ribeiro-Neto
   Modern Information Retrieval
2. Ian H. Witten et al.
   Managing Gigabytes: Compressing and Indexing Documents
3. Andreas M. Antonopoulos *(for semantic-blockchain integration topics, optional)*
4. Research papers from IEEE, ACM, Springer, Elsevier

Other useful resource(s):

1. Link to NPTEL course contents:https://onlinecourses.nptel.ac.in/noc26_cs13/preview
2. Link to NPTEL course contents:https://nptel.ac.in/courses/106106852

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Cloud Computing Essentials: Azure and AWS

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Java/Python. Basic Computer Fundamentals, Networking Fundamentals, Database Basics.

Course Objectives:

1. To build strong conceptual and practical understanding of cloud computing.
2. To enable hands-on expertise in AWS and Microsoft Azure core and advanced services.
3. To design scalable, secure, and fault-tolerant cloud architectures.
4. To implement networking, storage, compute, and database services.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To analyze cloud deployment and service models for real-world scenarios	Apply
CO-2	To deploy and manage AWS EC2, Azure VM, storage, and networking services	Understand
CO-3	Develop server less and event-driven applications. Use monitoring, logging, and auditing tools effectively.	Apply
CO-4	Evaluate automate infrastructure using Cloud Formation and ARM templates...	Analyze
CO-5	To understand and evaluate design multi-cloud and hybrid architectures.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Cloud Computing: Evolution from traditional IT to cloud, Characteristics: On-demand, elasticity, pay-as-you-go, Cloud service providers overview, Cloud Deployment Model, Public vs Private vs Hybrid vs Community cloud, Multi-cloud strategies , Use cases and decision factors, IaaS, PaaS, SaaS (deep comparison), Real-world examples, Responsibility sharing model, Virtualization & Containers, Hypervisors (Type 1 & Type 2), Virtual Machines vs Containers, Docker basics and container evolution.	**5**
2	AWS Global Infrastructure, Regions, Availability Zones, Edge locations, Fault isolation and resilience. AWS IAM, Users, Groups, Roles, Policies (JSON structure), IAM best practices (least privilege), MFA, API keys, credential management, EC2 (Compute Services), Instance types (General, Compute, Memory optimized), AMI, Launch templates , Key pairs and security groups, Pricing models (On-demand, Spot, Reserved, Dedicated), AWS Storage Basics, EBS volumes and types , EFS overview, EBS vs Instance Store.	**5**

3	AWS Advanced Services, Amazon S3, Buckets, Objects, metadata , Storage classes (Standard, IA, Glacier), Lifecycle policies , Versioning and replication, Static website hosting, Load Balancing & Auto Scaling, ELB types (CLB, ALB, NLB), Target groups, Health checks , Auto Scaling group, AWS Databases, RDS architecture and engines, Multi-AZ and Read replicas, Aurora advantages, DynamoDB (tables, partition key, indexing).	**6**
4	Caching (ElasticCache) , CDN & Edge Services, CloudFront working , Caching strategies, Signed URLs and cookies, Global Accelerato, Serverless Computing, AWS Lambda architecture, Event-driven computing, API Gateway integration, Cognito for authentication Virtual Private Cloud (VPC) Fundamentals, CIDR blocks, Subnetting (public/private), IP addressing, Internet Connectivity, Internet Gateway , NAT Gateway vs NAT Instance, Route tables, Security Mechanisms, Security Groups (stateful), NACL (stateless)	**5**
5	Best practices, Advanced Networking, VPC Peering, Endpoints (Interface & Gateway), Bastion host, Hybrid Connectivity, Site-to-Site VPN, Client VPN, Direct Connect, Route 53 routing policies, Health checks, VPC Flow Logs, Azure Fundamentals, Azure architecture, Subscriptions and resource groups, Regions and availability zones, Azure Active Directory, Users, Groups, Roles, RBAC vs IAM, MFA & Self-Service Password Reset	**6**
6	Privileged Identity Management, Azure Virtual Machines, VM sizes and series, Availability Sets & Zone, VM Scale Sets, VM images and extension, Azure Storage, Storage accounts , Blob storage (Hot, Cool, Archive), File, Queue, Table storage, Data redundancy (LRS, GRS, ZRS), Static website hosting	**5**
7	Azure Databases, Azure SQL Database, Azure MySQL, Cosmos DB (NoSQL models), Redis Cache, Azure Networking, Virtual Network (VNet), Subnets and IP addressing, NSG and ASG, Service endpoints . Load Balancing & DNS, Azure Load Balancer, Application Gateway (Layer 7), Azure DNS zones and records	**4**
8	ARM Templates, Template structure, Parameters, variables, outputs, Deployment strategies , Monitoring & Recovery, Azure Monitor and Logs, Azure Backup, Azure Site Recovery, DevOps & Automation, CI/CD pipelines, GitHub Actions, Azure DevOps, Terraform basics, GitOps principles Containers & Cloud Architecture, Docker fundamentals , Kubernetes (AKS basics), Microservices architecture, High availability & multi-region design	**6**
	Total lectures	**42**

Suggested Text Book(s):

1. Thomas Erl – Cloud Computing: Concepts, Technology & Architecture
2. Ben Piper – AWS Certified Solutions Architect Study Guide
3. Microsoft – Azure Fundamentals (AZ-900)
4. Michael Kavis – Architecting the Cloud

Suggested Reference Book(s):

1. Thomas Erl, Ricardo Puttini&Zaigham Mahmood – *Cloud Computing: Concepts, Technology & Architecture*.
2. Rajkumar Buyya, Christian Vecchiola& S. ThamaraiSelvi – *Mastering Cloud Computing*

Other useful resource(s):

1. Link to NPTEL course contents:

https://www.youtube.com/watch?v=NzZXz3fJf6o&list=PLShJJCRzJWxhz7SfG4hpaBD5bKOloWx9J

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Reinforcement Learning

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Probability & Statistics, Machine Learning

Course Objectives:

1. Describe the foundational concepts and historical evolution of reinforcement learning and its relevance to related fields.
2. Apply probability theory concepts such as random variables, distributions, and expectations in reinforcement learning contexts.
3. Model sequential decision-making problems using Markov Decision Processes and solve them using Bellman equations.
4. Implement model-free prediction and control algorithms including Monte Carlo and TD methods like SARSA and Q-Learning.
5. Apply function approximation techniques and experience replay mechanisms to enable scalable reinforcement learning.
6. Evaluate policy gradient methods and implement actor-critic algorithms, analyzing their bias-variance trade-offs.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Describe the foundational concepts and historical evolution of reinforcement learning and its relevance to related fields.	Remember, Understand
CO-2	Apply probability theory concepts such as random variables, distributions, and expectations in reinforcement learning contexts.	Apply
CO-3	Model sequential decision-making problems using Markov Decision Processes and solve them using Bellman equations.	Apply
CO-4	Implement model-free prediction and control algorithms including Monte Carlo and TD methods like SARSA and Q-Learning.	Apply
CO-5	Apply function approximation techniques and experience replay mechanisms to enable scalable reinforcement learning.	Apply
CO-6	Evaluate policy gradient methods and implement actor-critic algorithms, analyzing their bias-variance trade-offs.	Evaluate

Course Contents:

Unit	Contents	Lectures
1	Introduction: Origin and history of Reinforcement Learning research. Its connections with other related fields and with different branches of machine learning.	**2**
2	Probability Primer: Axioms of probability, concepts of random variables, PMF, PDFs, CDFs, Expectation. Concepts of joint and multiple random variables, joint,	**4**

	conditional and marginal distributions. Correlation and independence.	
3	Markov Decision Process: Introduction to RL terminology, Markov property, Markov chains, Markov reward process (MRP). Introduction to and proof of Bellman equations for MRPs along with proof of existence of solution to Bellman equations in MRP. Introduction to Markov decision process (MDP), state and action value functions, Bellman expectation equations, optimality of value functions and policies, Bellman optimality equations.	**6**
4	Prediction & Control by Dynamic Programming: Overview of dynamic programming for MDP, definition and formulation of planning in MDPs, principle of optimality, iterative policy evaluation, policy iteration, value iteration, Banach fixed point theorem, proof of contraction mapping property of Bellman expectation and optimality operators, proof of convergence of policy evaluation and value iteration algorithms, DP extensions.	**6**
5	Monte Carlo methods for Model Free Prediction & Control: Overview of Monte Carlo methods for model free RL, First visit and every visit Monte Carlo, Monte Carlo control, On policy and off policy learning, Importance sampling.	**6**
6	TD Methods: Incremental Monte Carlo Methods for Model Free Prediction, Overview TD (0), TD (1) and TD (λ), k-step estimators, unified view of DP, MC and TD evaluation methods, TD Control methods - SARSA, Q-Learning and their variants.	**6**
7	Function Approximation Methods: Getting started with the function approximation methods, Revisiting risk minimization, gradient descent from Machine Learning, Gradient MC and Semi-gradient TD (0) algorithms, Eligibility trace for function approximation, After states, Control with function approximation, Least squares, Experience replay in deep Q-Networks.	**6**
8	Policy Gradients: Getting started with policy gradient methods, Log derivative trick, Naive REINFORCE algorithm, bias and variance in Reinforcement Learning, Reducing variance in policy gradient estimates, baselines, advantage function, actor-critic methods.	**6**
	Total lectures	**42**

Suggested Text Book(s):

1. Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. 2nd Edition, MIT Press, 2018.
2. Alberto Leon-Garcia. Probability, Statistics, and Random Processes for Electrical Engineering. 3rd Edition, Pearson Education, 2008.

Suggested Reference Book(s):

1. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106106143
2. Link to topics related to course:
   https://www.youtube.com/watch?v=JgvyzIkgxF0
   https://www.youtube.com/watch?v=LzaWrmKL1Z4

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	3	2	2	2	2	2	3	**2.18**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	**1.73**
CO-4	2	3	3	3	3	1	1	1	2	3	2	**2.18**
CO-5	2	2	2	1	3	2	1	1	2	1	1	**1.64**
CO-6	2	2	2	2	3	2	2	2	2	2	2	**2.09**
Average	**2**	**2.33**	**2.33**	**2.17**	**3**	**1.5**	**1.33**	**1.33**	**2**	**2**	**1.67**	

Responsible AI and AI Ethics

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Artificial Intelligence

Course Objectives:

1. Study the concepts of morality and ethics in Artificial Intelligence.
2. Learn about ethical initiatives in the field of Artificial Intelligence.
3. Understand AI standards, regulations, and ethical frameworks.
4. Study the social and ethical issues related to Robotics.
5. Explore challenges and opportunities in AI ethics with national and international strategies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Developed foundational understanding of Responsible AI principles and AI ethics	Understand
CO-2	Acquired knowledge of real-time ethical issues, challenges, and risks in AI applications.	Evaluate
CO-3	Understood ethical harms, biases, and global ethical initiatives for Responsible AI.	Evaluate
CO-4	Learned about AI governance, standards, and regulations for designing safe and trustworthy AI systems.	Apply
CO-5	Understood concepts of Roboethics, morality, and professional responsibility in robotics and AI systems	Understand

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Responsible AI:** Definition and scope of Responsible AI and AI Ethics, Importance of Responsible AI in society, Principles of Responsible AI (fairness, accountability, transparency, privacy, safety, trust), Human intelligence vs machine decision-making, Impact of AI on society, Impact on human psychology, Impact on legal systems, Impact on environment and sustainability, Trust worthiness and reliability of AI systems.	**8**
2	**Ethical initiatives in AI:** International ethical initiatives and guidelines, Ethical harms and concerns in AI systems, Bias and discrimination in AI, Privacy and surveillance issues, Ethical risks in AI deployment, Case studies on ethical issues in healthcare AI systems, Autonomous vehicles, Warfare and weaponization, Role of responsible innovation and ethical risk management.	**8**

3	**AI standards and regulations:** Model process for addressing ethical concerns during AI system design, Transparency and explainability of AI systems, Data privacy process and protection techniques, Algorithmic bias considerations and mitigation, Accountability and auditability in AI models, Ontological standards for ethically driven robotics and automation systems, AI governance frameworks, National and international AI regulations.	9
4	**Roboethics:** Introduction to robotics and intelligent robots, Concepts of Roboethics, Ethics and morality in robotics, Moral theories applied to robotics, Ethics in science and technology, Ethical issues in ICT society, Harmonization of ethical principles in robotics, Ethics and professional responsibility in robotics development, Roboethics taxonomy and classification.	9
5	**Responsible AI challenges and opportunities:** Challenges in implementing Responsible AI, Opportunities created by ethical AI systems, Ethical issues in artificial intelligence applications, Societal issues concerning the application of AI in medicine, AI-based decision-making role in industries, Trust and safety challenges in autonomous and semi-autonomous systems, National and international strategies on Responsible AI, Future trends and challenges in AI ethics.	8
Total lectures		**42**

Suggested TextBook(s):

1. Eleanor Bird, Jasmin Fox-Skelly, Nicola Jenner, Ruth Larbey, Emma Weitkamp and Alan Winfield, **"The Ethics of Artificial Intelligence: Issues and Initiatives"**, European Parliamentary Research Service Scientific Foresight Unit (STOA), PE 634.452, March 2020.
2. Patrick Lin, Keith Abney, George A Bekey, **"Robot Ethics: The Ethical and Social Implications of Robotics"**, The MIT Press, January 2014.
3. Dignum, V., Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way, Springer Nature (2023).

Suggested Reference Book(s):

1. Paula Boddington, "Towards a Code of Ethics for Artificial Intelligence (Artificial Intelligence: Foundations, Theory, and Algorithms)", November 2017.
2. Mark Coeckelbergh, **"AI Ethics"**, The MIT Press Essential Knowledge Series, April 2020.

Other useful resource(s):

1. https://sci-hub.mksa.top/10.1007/978-3-540-30301-5_65
2. https://www.scu.edu/ethics/all-about-ethics/artificial-intelligence-and-ethics-sixteen-challenges-and-opportunities/
3. https://www.weforum.org/agenda/2016/10/top-10-ethical-issues-in-artificial-intelligence/
4. https://sci-hub.mksa.top/10.1159/000492428

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	1	1	2	1	1	1	1	2	1	1	1.2
CO-2	2	1	1	3	1	1	2	1	1	3	1	1.4
CO-3	2	1	1	3	1	1	2	1	1	3	2	1.5
CO-4	2	2	1	1	2	3	2	2	4	3	1	2.0
CO-5	2	2	3	1	2	1	3	1	1	3	3	2.0
Average	2	2	3	3	1	3	3	1	3	2	2	2.2

Machine Learning and Big Data

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basics of Machine Learning, Python Programming, Basic NLP

Course Objectives:

1. To understand machine learning and big data concepts
2. To study distributed architecture for big data processing.
3. To study and apply scalable machine learning algorithms for big data.
4. To implement supervised and unsupervised algorithms for scalable solutions.
5. To design a recommendation system for real life NLP dataset.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain machine learning and big data concepts	Understand
CO-2	Apply data science fundamentals to big data	Apply
CO-3	Implement machine learning algorithms using Apache Spark	Apply
CO-4	Analyze supervised and unsupervised learning algorithms	Analyze
CO-5	Design recommendation system using MLlib library for NLP dataset	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Machine Learning:** Fundamentals and Definition of Machine Learning, Types of Machine Learning, Learning of Models Based on the First Criteria, Learning of Models Based on Second Criteria (Batch Mode Learning and Online Mode Learning), Applications of Machine Learning	6
2	**Introduction to Big Data:** Big Data: Why and Where, Characteristics of Big Data and Dimensions of Scalability, Data Science: Getting Value out of Big Data, Foundations for Big Data Systems and Programming	3
3	**Apache Spark:** Core Concepts, Spark Components, Architecture, working of Spark, Apache Spark Environment Setup and Configuration: Installation on Linux, Mac and Windows. Standalone setup and local cluster setup	5

4	**Apache Spark MLlib:** Programming with RDDs, Statistics and Linear Algebra Preliminaries,Data Exploration: Univariate Analysis & Multivariate Analysis, Data Manipulation: Feature Extraction, Feature Transform, Feature Selection, Unbalanced data: Under sampling, Regression: Linear Regression, Generalized linear regression, Decision tree Regression, Random Forest Regression, Gradient-boosted tree regression,Regularization: Ordinary least squares regression, Ridge regression, Least Absolute Shrinkage and Selection Operator (LASSO), Elastic net	**9**
5	**Supervised and Un-supervised Learning with Spark:** Classification: Binomial logistic regression, Multinomial logistic regression, Decision tree Classification, Random Forest Classification, Gradient-boosted tree Classification, XGBoost: Gradient-boosted tree Classification, Naive Bayes Classification	**10**
6	**Natural Language Processing with Apache Spark:** Text Mining: Text Collection, Text Preprocessing, Text Classification, Sentiment analysis, N-grams and Correlations, Topic Model: Latent Dirichlet Allocation	**6**
7	**Recommendation Engine with Distributed Framework:** Recommender systems: Alternating Least Squares	**3**
Total lectures		**42**

Suggested Text Book(s):

1. **Machine Learning and Big Data: Concepts, Algorithms, Tools and Applications**

Editor(s):Uma N. Dulhare, Khaleel Ahmad, KhairolAmali Bin Ahmad
(Publisher: Scrivener Publishing LLC)
(Print ISBN:9781119654742 |Online ISBN:9781119654834)
https://onlinelibrary.wiley.com/doi/book/10.1002/9781119654834

2. **Practical Machine Learning with Spark**

Gourav Gupta, Dr. Manish Gupta, Dr. Inder Singh Gupta
(BPB Publisher) (ISBN: 9789391392086, eISBN: 9789391392130)
https://in.bpbonline.com/products/practical-machine-learning-with-spark

Suggested Reference Book(s):

1. Spark: The Definitive Guide, Big Data Processing Made Simple

Bill Chambers and MateiZaharia
(Publisher: O'Reilly)(ISBN: 9781491912201)
https://www.oreilly.com/library/view/spark-the-definitive/9781491912201/

2. Essential PySpark for Scalable Data Analytics: A beginner's guide to harnessing the power and ease of PySpark 3

SreeramNudurupati
(Publisher: packtpublications)(ISBN-13 :9781800568877)
https://www.packtpub.com/en-ch/product/essential-pyspark-for-scalable-data-analytics-9781800568877

Other useful resource(s):

1. Coursera: Big Data Specialization: UCSanDiego

https://www.coursera.org/specializations/big-data

2. Coursera: Machine Learning with PySpark: Edureka
https://www.coursera.org/learn/machine-learning-with-pyspark
3. Coursera:Machine Learning with Apache Spark: IBMhttps://www.coursera.org/learn/machine-learning-with-apache-spark
4. Coursera: Introduction to Big Data with Spark and Hadoop: IBM
https://www.coursera.org/learn/introduction-to-big-data-with-spark-hadoop

Evaluation Scheme:

S. No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (Pos)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	2	2	1	1	1	1	1	1	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	2	1	**2.0**
CO-3	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-4	2	3	2	3	2	1	1	1	2	2	2	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.6**	**2.2**	**2.2**	**2.6**	**1.0**	**1.0**	**1.2**	**2.0**	**1.8**	**1.6**	

Secure Design of Software Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Programming Fundamentals, Data Structures, Software Engineering

Course Objectives:

1. To understand fundamental principles of secure software design and security engineering practices.
2. To analyze common software vulnerabilities and attack vectors in modern software systems.
3. To apply secure design principles, threat modeling, and risk assessment techniques.
4. To design software architectures incorporating confidentiality, integrity, availability, and privacy requirements.
5. To implement secure coding practices and verification techniques throughout the Software Development Life Cycle (SDLC).

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain core concepts of software security, security goals, and threat landscape.	Understand
CO-2	Apply secure design principles and threat modeling techniques to software systems.	Apply
CO-3	Analyze software vulnerabilities and security risks using systematic approaches.	Analyze
CO-4	Evaluate secure architectures and defense mechanisms against common attacks.	Evaluate
CO-5	Design secure software systems integrating secure coding and testing practices.	Create

Course Contents:

Unit	Contents	Lectures Required
1	**Introduction to Software Security:** Security concepts, CIA triad, security goals, secure software engineering, security vs reliability, overview of cyber threats, security design lifecycle, principles of least privilege and defense in depth.	**5**
2	**Secure Software Development Life Cycle (SSDLC):** Security requirements engineering, risk assessment, security policies, secure design review, DevSecOps concepts, integration of security in Agile and DevOps models.	**6**
3	**Threat Modeling and Risk Analysis:** Risk Management Life Cycle, Risk Profiling, Risk Exposure Factors, STRIDE and DREAD models, attack surface analysis, misuse/abuse cases, adversarial thinking, threat identification and mitigation strategies.	**6**

4	**Secure Design Principles and Architecture:** Secure architectural patterns, trust boundaries, secure interfaces, authentication and authorization models, access control mechanisms, secure microservices and distributed systems design.	**7**
5	**Common Software Vulnerabilities:** OWASP Top 10, buffer overflow, injection attacks (SQL/Command), cross-site scripting (XSS), cross-site request forgery (CSRF), insecure deserialization, race conditions.	**7**
6	**Secure Coding Practices:** Input validation, error handling, secure API design, cryptographic best practices, secrets management, code review techniques, static and dynamic analysis tools.	**6**
7	**Security Testing and Verification:** Secure Software Development Life Cycle, Risk Based Security Testing, Prioritizing Security Testing With Threat Modeling, Penetration testing basics, fuzz testing, vulnerability assessment, security auditing, automated testing tools, secure deployment and monitoring practices.	**5**
Total lectures		**42**

Suggested Text Book(s)

1. Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley, 3rd Edition.
2. Gary McGraw, Software Security: Building Security In, Addison-Wesley.
3. Mark Dowd, John McDonald & Justin Schuh, The Art of Software Security Assessment, Addison-Wesley.
4. Julia H. Allen et al., Software Security Engineering, SEI Series, Addison-Wesley.

Suggested Reference Book(s)

1. Michael Howard & David LeBlanc, Writing Secure Code, Microsoft Press.
2. Adam Shostack, Threat Modeling: Designing for Security, Wiley. OWASP Foundation Documentation (Latest Edition). Viega& McGraw, Building Secure Software, Addison-Wesley.

Other useful resource(s):

1. Link to NPTEL course contents:Software Security Courses:https://www.youtube.com/watch?v=2xrsRlWmcRQ
2. Link to NPTEL course contents:https://onlinecourses-archive.nptel.ac.in/noc19_cs24/preview
3. Link to topics related to course:
   - OWASP Secure Coding Practices Guide
   - https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/
   - https://www.youtube.com/watch?v=teLmHBPGiV8

Evaluation Scheme:

S. No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	2	1	1	2	**1.8**
CO-2	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-3	3	3	2	3	3	1	1	2	2	2	2	**2.2**
CO-4	2	3	3	3	3	1	1	2	2	2	3	**2.3**
CO-5	3	2	3	2	3	1	1	3	3	2	2	**2.3**
Average	**2.8**	**2.6**	**2.6**	**2.2**	**2.8**	**1.0**	**1.0**	**2.2**	**2.0**	**1.8**	**2.2**	

Fog and Edge Computing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Cloud Computing, Computer Networks

Course Objectives:

1. To introduce the fundamental concepts of IoT, cloud computing, and fog and edge computing paradigms.
2. To understand architectures, integration models, and challenges in fog and edge computing environments.
3. To develop knowledge of middleware design, data management, and resource allocation techniques in fog systems.
4. To explore security, privacy, machine learning, and optimization techniques in IoT and fog computing.
5. To analyze and design real-world applications of fog and edge computing using simulation tools and case studies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the fundamental concepts of IoT, cloud computing, and the need for fog and edge computing paradigms.	Understand
CO-2	Analyze architectures, hierarchies, and integration models of IoT, fog, and cloud systems.	Analyze
CO-3	Apply middleware concepts and data management techniques for efficient fog and edge computing environments.	Apply
CO-4	Evaluate security challenges, privacy issues, and machine learning approaches in IoT and fog systems.	Analyze
CO-5	Analyze optimization techniques, performance metrics, and resource management strategies in fog computing.	Analyze
CO-6	Design and assess real-world applications of fog and edge computing using simulation tools and case studies.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Foundations of IoT, Fog & Edge Computing** Introduction to IoT and emerging computing paradigms Cloud-centric IoT (CIoT) and its limitations Need for Fog and Edge Computing Comparison of Cloud, Fog, Edge, and Mist computing Architecture and hierarchy of Fog and Edge Computing Advantages (SCALE: Security, Cognition, Agility, Latency, Efficiency) Core mechanisms (SCANC: Storage, Compute, Acceleration,	10

	Networking, Control) Business models (XaaS, application and support services) Opportunities and research challenges	
2	**Architecture, Integration & Challenges** Federated edge systems and architecturesNetworking challenges in edge environmentsResource discovery, management, and service migrationLoad balancing and mobility supportIntegration of IoT, Fog, and Cloud (C2F2T model)Modeling techniques such as Analytical models, Petri Nets, and ILPRole of 5G in edge computingNetwork slicing and orchestration	**8**
3	**Middleware & Data Management** Need and role of middleware in fog and edge computing, Middleware design goals and architecture, Device discovery and context-awareness Scheduling, resource allocation, and execution management, Container-based edge computing using Docker and lightweight clustersFog data lifecycle and management, Data preprocessing including cleaning, fusion, and edge analytics, Data storage, placement, and privacy	**8**
4	**Security, Machine Learning & Optimization** Security challenges in IoT and fog environments, Privacy issues and attack models such as DDoS and malware, Machine learning for IoT security, Machine learning techniques including classification, clustering, and ANN, Optimization problems in fog computing, Performance metrics including QoS, energy, cost, and resource utilization	**6**
5	**Applications & Case Studies** Fog computing for big data analytics, Smart home and smart city applications, Healthcare monitoring systems, Smart surveillance and video analytics, Intelligent transportation systems, Testing perspectives of fog-based IoT applications	**6**
6	**Simulation, Legal & Emerging Topics** Simulation tools such as iFogSim and modeling of fog environments Case studies and performance evaluation, Legal and regulatory issues including data protection and privacy laws, Future research directions and emerging trends	**4**
	Total lectures	**42**

Suggested Text Book(s):

1. Fog and Edge Computing: Principles and Paradigms by Rajkumar Buyya and Satish Narayana Srirama, Wiley, 2019
2. Edge Computing: Vision and Challenges by Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu, IEEE Internet of Things Journal
3. Cloud Computing: Concepts, Technology & Architecture by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, Pearson

Suggested Reference Book(s):

1. Fog Computing in the Internet of Things: Intelligence at the Edge by Amir M. Rahmani, PasiLiljeberg, Jürgo-SörenPreden, and Axel Jantsch, Springer, 2018
2. Fog/Edge Computing for Security, Privacy, and Applications edited by Wei Chang and Jie Wu, Springer, 2021
3. Energy Conservation Solutions for Fog-Edge Computing Paradigms edited by Rajeev Tiwari, Mamta Mittal, and Lalit Mohan Goyal, Springer, 2022
4. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola, and S. ThamaraiSelvi, McGraw Hill

Other useful resource(s):

1. NPTEL Online Course on Cloud Computing and Distributed Systems (IIT lectures)
2. IEEE Xplore Digital Library for latest research papers on fog and edge computing
3. OpenFog Consortium architecture and white papers on fog computing
4. ETSI resources on Multi-access Edge Computing (MEC)
5. iFogSim toolkit for modeling and simulation of fog environments
6. GitHub repositories and open-source platforms for IoT and edge computing implementations

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.63**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.09**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.1**
CO-6	2	2	2	2	2	1	1	1	3	2	2	**1.81**
Average	**2.6**	**2.3**	**2.3**	**2**	**2.5**	**1**	**1**	**1.1**	**2.3**	**1.8**	**2.1**	

Statistical Analysis and Computing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: Specialization Core

L-T-P: 3-0-0

Pre-requisite: Basics of probability and statistics

Course Objectives:

1. To introduce basic concepts of descriptive statistics and probability.
2. To develop understanding of statistical inference methods for data analysis.
3. To enable effective data visualization using standard graphical tools.
4. To apply data transformation and dimensionality reduction techniques.
5. To provide an overview of supervised and unsupervised learning methods.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain key concepts of descriptive statistics, probability theory, and probability distributions.	Understand
CO-2	Apply statistical inference techniques such as confidence intervals and hypothesis testing to real data problems.	Understand
CO-3	Create and interpret visualizations including box plots, scatter plots, and Pareto charts for exploratory analysis.	Evaluate
CO-4	Implement data transformation and dimensionality reduction methods.	Evaluate
CO-5	Analyze basic supervised and unsupervised learning models and assess their performance on datasets.	Apply

Course Contents:

Unit	Contents	Lectures required
1	Descriptive Statistics and Probability: Mean, Median, Mode, Variance, Standard Deviation, Probability axioms, Conditional probability, Bayes theorem, skewness, kurtosis, Probability Distributions: Binomial, Poisson, Normal, Exponential, Central Limit Theorem	**9**
2	Statistical Inference: Point and interval estimation, Confidence intervals, Hypothesis testing – Z, T, Chi-square, ANOVA	**9**
3	Data Visualization and Exploration: Box Plot, Histogram, Multi-Vari Chart, Run Chart, Pareto Chart, Scatter Plot, Stem-and-leaf plot.	**4**
4	Data transformation, dimensionality reduction: Power transformation, standardization, multidimensional scaling, principal component analysis, linear or non-linear dimensionality reduction.	**7**

5	Introduction to Supervised and Unsupervised Learning: Goals and applications of machine learning, classification methods: Linear Regression, Logistic Regression, DT, SVM, KNN, Naïve Bayes, Data clustering methods: k-means, Hierarchal methods.	**9**
6	Smoothing Techniques and Visual Analytics in Statistics: Bivariate distribution smooths, curve fitting toolbox, Dendrogram, tree maps, rectangle plots.	**4**
Total lectures		**42**

Suggested Text Book(s):

1. Walpole, R.E., Myers, R.H., Myers, S.L., & Ye, K., Probability and Statistics for Engineers and Scientists, Pearson.
2. Montgomery, D.C., &Runger, G.C., Applied Statistics and Probability for Engineers, Wiley.
3. Tom Mitchell, “Machine Learning”, McGraw Hill, 1997, ISBN 0070428077.

Suggested Reference Book(s):

1. Ross, S.M., Introduction to Probability and Statistics for Engineers and Scientists, Academic Press.
2. Kutner, M.H., Nachtsheim, C.J., Neter, J., & Li, W., Applied Linear Statistical Models, McGraw-Hill Education.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/111105041
2. https://www.edx.org/learn/machine-learning

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	2	2	1	1	1	2	1	2	**1.64**
CO-2	3	3	2	2	2	1	1	1	2	1	2	**1.82**
CO-3	2	2	2	2	2	1	1	1	3	2	2	**1.82**
CO-4	2	2	2	3	3	1	1	1	2	2	2	**1.91**
CO-5	2	3	3	3	3	2	1	1	2	2	3	**2.27**
Average	**2.4**	**2.4**	**2.0**	**2.4**	**2.4**	**1.2**	**1.0**	**1.0**	**2.2**	**1.6**	**2.2**	

Time Series Analysis and Forecasting

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basics of Statistics

Course Objectives:

1. To introduce the principles and methods of forecasting.
2. To introduce various components of time series and time series models which cater to the real-world
3. To help students explore and use the various criteria used for performance evaluation.
4. To address both the aspects of descriptive and predictive analytics.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the principles and process of forecasting.	Understand
CO-2	Apply and analyze univariate ARIMA methods for real world problems.	Apply
CO-3	Apply and analyze smoothing methods for real world problems.	Apply
CO-4	Apply various criteria for evaluating model quality	Apply
CO-5	Apply and analyze multivariate methods for real world problems.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Foundations of Forecasting: Planning and Forecasting, Statistical Foundations for Forecasting, Simple Linear Regression Analysis	**5**
2	Univariate Smoothing Methods: Simple Smoothing Methods, Decomposition Methods and Seasonal Indexes, Trend-Seasonal and Holt-Winters Smoothing	**7**
3	Univariate ARIMA Methods: Univariate ARIMA Models: Introduction ARIMA Applications, ARIMA Forecast Intervals	**7**
4	Multivariate / Causal Methods: Multiple Regression of Time Series Econometric Methods, ARIMA Intervention Analysis, Multivariate ARIMA Transfer Functions	**9**
5	Cyclical, Qualitative & AI Methods: Cyclical Forecasting Methods Technological & Qualitative Forecasting (Long-Term Forecasting) Artificial Neural Networks, Expert Systems & Genetic Algorithms	**7**

6	Combining & Evaluation Methods: Control, Validation & Combining Methods, Method Characteristics, Accuracy & Data Sources	**7**
Total lectures		**42**

Suggested Text Book(s):

1. Stephen A. DeLurgio, Forecasting Principles and Applications, McGraw Hill International Editions 2 Revised ed Edition – 1 July 1998
2. Rob J Hyndman and George Athanasopoulos, Forecasting: Principles and Practice, Third Edition, Otexts; 2018

Suggested Reference Book(s):

1. Ruet S. Tsay, Analysis of Financial Time Series, 3rd Edition, Wiley, New Jersy, 2015
2. Walter Enders, Applied Econometrics, 3rd Edition, Wiley, New Jersy, 2014
3. Terence C. Mills, The Foundations of Modern Time Series Analysis, Palgrave Macmillan, 2011

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106101578
2. Link to NPTEL course contents:https://nptel.ac.in/courses/103106123

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Edge and Federated Learning

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning, Deep Learning, Python Programming, Basics of Computer Networks.

Course Objectives:

1. To understand the architecture and motivation of edge computing and federated learning for privacy-preserving, low-latency AI systems.
2. To explore model compression and optimisation techniques (pruning, quantisation, distillation) for deployment on resource-constrained edge devices.
3. To study federated learning algorithms and address challenges of statistical and system heterogeneity in distributed settings.
4. To analyse communication efficiency challenges and evaluate gradient compression and scheduling strategies in federated learning.
5. To analyse privacy and security threats in FL and evaluate defence mechanisms including differential privacy and secure aggregation

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of edge computing, federated learning, and the motivations behind decentralised AI, including privacy, latency, and bandwidth constraints.	Understand
CO-2	Explain model compression and optimisation techniques — pruning, quantisation, and knowledge distillation — for deploying neural networks on resource-constrained edge devices.	Understand
CO-3	Apply federated learning algorithms (FedAvg, FedProx) to distributed ML settings and address challenges of statistical and system heterogeneity.	Apply
CO-4	Analyse communication efficiency challenges in federated learning and evaluate strategies including gradient compression, local SGD, and bandwidth-aware client selection.	Analyze
CO-5	Analyse privacy and security threats in federated learning and evaluate defence mechanisms including differential privacy, secure aggregation, and Byzantine-robust aggregation.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Edge AI and Federated Learning: Limitations of centralised cloud ML; motivation for edge computing and on-device inference; overview of federated learning — key concepts, taxonomy, and real-world use cases (healthcare, IoT, mobile); comparison of horizontal FL, vertical FL, and federated transfer learning.	4

2	Edge Computing Fundamentals: Edge vs. fog vs. cloud computing; edge hardware — microcontrollers, SoCs, GPUs (NVIDIA Jetson), TPUs, FPGAs; model deployment constraints (memory, power, latency); TinyML and embedded inference; edge inference frameworks — TensorFlow Lite, ONNX Runtime, OpenVINO.	**5**
3	Model Compression and Optimisation for Edge: Pruning — structured and unstructured; quantisation — post-training quantisation, quantisation-aware training; knowledge distillation; neural architecture search (NAS) for edge; benchmarking edge models.	**5**
4	Federated Learning Algorithms: The FedAvg algorithm — convergence analysis; FedProx for heterogeneous systems; personalised FL — local fine-tuning, meta-learning (MAML); asynchronous FL; hierarchical FL; client selection and scheduling strategies.	**7**
5	Data and System Heterogeneity: Non-IID data distributions — label skew, feature skew, quantity skew; impact on model convergence; techniques to handle non-IID data — data augmentation, clustering, multi-task FL; system heterogeneity — stragglers, partial participation, fault tolerance.	**6**
6	Communication Efficiency in Federated Learning: Communication bottlenecks in FL; gradient compression — sparsification and quantisation; sketching and randomised methods; local SGD and lazy aggregation; gossip-based and peer-to-peer FL; bandwidth-aware client selection.	**6**
7	Privacy, Security, Frameworks and Deployment: Privacy threats — gradient inversion, membership inference, model inversion attacks; differential privacy — local DP and central DP, Gaussian mechanism, DP-SGD; secure aggregation protocols; homomorphic encryption basics; Byzantine robustness — Krum, coordinate-wise median, FLTrust; FL frameworks — PySyft, Flower (flwr), TensorFlow Federated (TFF), FedML; cross-device vs. cross-silo FL; case studies — Google Gboard, Apple on-device learning, healthcare FL; ethical and regulatory considerations (GDPR, HIPAA).	**9**
	Total lectures	**42**

Suggested Text Book(s):

1. McMahan, H. B., et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS, 2017. (Foundational FL paper)
2. Warden, P. and Situnayake, D., TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2020.
3. Konečný, J. et al., Federated Learning: Strategies for Improving Communication Efficiency, NIPS Workshop, 2016.
4. Li, T. et al., Federated Learning: Challenges, Methods, and Future Directions, IEEE Signal Processing Magazine, 2020.

Suggested Reference Book(s):

1. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Shi, W. et al., Edge Computing: Vision and Challenges, IEEE Internet of Things Journal, 2016.
3. Dwork, C. and Roth, A., The Algorithmic Foundations of Differential Privacy, Foundations and Trends in Theoretical Computer Science, 2014.

Other useful resource(s):

1. TensorFlow Federated (TFF): https://www.tensorflow.org/federated
2. Flower Federated Learning Framework: https://flower.dev
3. PySyft (OpenMined): https://github.com/OpenMined/PySyft

4. NPTEL – Privacy in the age of AI: https://nptel.ac.in
5. Papers With Code – Federated Learning: https://paperswithcode.com/task/federated-learning
6. CMU Federated Learning Course Notes: https://cmu.edu

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	2	2	2	2	1	1	1	2	1	2	**1.7**
CO-3	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-4	3	3	2	3	2	1	1	1	2	2	2	**2.0**
CO-5	3	2	3	3	2	2	1	2	2	2	2	**2.1**
Average	**3.0**	**2.4**	**2.4**	**2.2**	**2.2**	**1.2**	**1.0**	**1.2**	**2.0**	**1.6**	**2.0**	

Retrieval Augmented Generation

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning, NLP

Course Objectives:

1. Understand LLM limitations and RAG needs
2. Study embeddings and retrieval
3. Implement RAG pipelines
4. Evaluate RAG systems
5. Design domain-specific applications

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain RAG fundamentals	Understand
CO-2	Apply retrieval techniques	Apply
CO-3	Implement RAG pipelines	Apply
CO-4	Analyze RAG systems	Analyze
CO-5	Design RAG systems	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to RAG: LLM limitations, hallucination, knowledge grounding, RAG architecture	**4**
2	Text Representation: Embeddings (Word2Vec, BERT, OpenAI embeddings), similarity metrics	**6**
3	Vector Databases: FAISS, Pinecone, indexing, ANN search, chunking strategies	**6**
4	Retrieval Techniques: Dense vs sparse retrieval, hybrid search, reranking	**6**
5	RAG Pipeline Design: Query processing, retrieval, augmentation, generation workflows	**7**
6	Advanced RAG: Multi-hop retrieval, query rewriting, agentic RAG, graph-based retrieval	**5**

7	Evaluation of RAG Systems: Metrics (precision, recall, faithfulness), benchmarking	**4**
8	Applications: QA systems, chatbots, enterprise search, domain-specific assistants	**4**
Total lectures		**42**

Suggested Text Book(s):

1. AI Engineering — Chip Huyen
2. Introduction to Information Retrieval — Christopher D. Manning

Suggested Reference Book(s):

1. Speech and Language Processing — Dan Jurafsky& James H. Martin
2. Ian Goodfellow, *Deep Learning*, MIT Press

Other useful resource(s):

1. Udemy https://www.coursera.org/professional-certificates/ibm-rag-and-agentic-ai
2. Coursera: https://www.coursera.org/professional-certificates/ibm-rag-and-agentic-ai
3. NPTEL: https://www.youtube.com/watch?v=-DGMjrOiyO8&list=PLp6ek2hDcoNDDRINFiWGDlPKUwW-g1Hjk

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	2	2	1	1	1	1	1	1	1.5
CO-2	3	3	2	2	3	1	1	1	2	2	1	2.0
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	2	3	2	3	2	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	2	3	2	2	2.2
Average	2.8	2.6	2.2	2.2	2.6	1.0	1.0	1.2	2.0	1.8	1.6	

Machine Learning and Natural Language Processing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Probability and Statistics, Linear Algebra, Python Programming, Data Structures and Algorithms.

Course Objectives:

1. To introduce the principles, algorithms, and mathematical foundations of supervised and unsupervised machine learning.
2. To develop understanding of ensemble methods, model selection, regularisation, and performance evaluation techniques.
3. To familiarise students with the linguistic foundations of NLP and classical text processing pipelines including language models, POS tagging, and parsing.
4. To build understanding of deep learning architectures — RNNs, LSTMs, and Transformers — and their application to NLP tasks.
5. To explore state-of-the-art pre-trained language models (BERT, GPT, T5) and analyse their application to downstream NLP tasks.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the foundational concepts of machine learning including supervised, unsupervised, and reinforcement learning paradigms, along with bias-variance trade-off and model evaluation techniques.	Understand
CO-2	Explain core NLP concepts including text preprocessing, language models, POS tagging, parsing, and sentiment analysis using both statistical and neural approaches.	Understand
CO-3	Apply classical ML algorithms — linear/logistic regression, SVMs, decision trees, and ensemble methods — to solve real-world structured and text-based problems.	Apply
CO-4	Apply deep learning architectures — RNNs, LSTMs, and Transformers — and pre-trained language models (BERT, GPT, T5) to NLP tasks such as classification, NER, and machine translation.	Apply
CO-5	Analyse challenges in ML and NLP pipelines including statistical heterogeneity, model selection, hyperparameter tuning, and performance interpretation on benchmark datasets.	Analyze

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Machine Learning: Types of ML — supervised, unsupervised, semi-supervised, reinforcement learning; the ML pipeline — data collection, preprocessing, feature engineering, model training, evaluation, deployment; bias-variance trade-off; overfitting and underfitting; cross-validation; performance metrics — accuracy, precision, recall, F1, ROC-AUC.	**4**

2	Supervised Learning Algorithms: Linear regression — OLS, gradient descent, regularisation (Ridge, Lasso); logistic regression — binary and multiclass; Support Vector Machines — hard and soft margin, kernel trick; k-Nearest Neighbours; Naive Bayes classifier; decision trees — ID3, C4.5, CART; ensemble methods — bagging, random forests, boosting (AdaBoost, Gradient Boosting, XGBoost).	**9**
3	Unsupervised Learning and Dimensionality Reduction: K-Means clustering — algorithm, elbow method; hierarchical clustering — agglomerative, dendrogram; DBSCAN; Gaussian Mixture Models and EM algorithm; dimensionality reduction — PCA, t-SNE; anomaly detection.	**5**
4	Introduction to Natural Language Processing: Text as data — challenges and pipeline; text preprocessing — tokenisation, stop-word removal, stemming, lemmatisation; text representation — Bag of Words, TF-IDF, n-grams; word embeddings — Word2Vec (CBOW, Skip-gram), GloVe, FastText; introduction to NLTK and spaCy.	**5**
5	Core NLP Tasks — Statistical and Classical Approaches: Language models — N-gram models, smoothing techniques; Part-of-Speech (POS) tagging — HMM-based tagging, Viterbi algorithm; Named Entity Recognition (NER) — rule-based and CRF-based; syntactic parsing — constituency and dependency parsing; sentiment analysis — lexicon-based and ML-based approaches; topic modelling — LDA.	**7**
6	Deep Learning for NLP: Recurrent Neural Networks (RNNs) — vanishing gradient problem; Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU); sequence-to-sequence models with attention mechanism; Convolutional Neural Networks for text classification; Transformer architecture — self-attention, multi-head attention, positional encoding; encoder-decoder structure.	**6**
7	Pre-trained Language Models and Transfer Learning: BERT — pre-training objectives (MLM, NSP), fine-tuning for downstream tasks; GPT — autoregressive language modelling; RoBERTa, DistilBERT; T5 and BART for seq2seq tasks; prompt engineering and in-context learning; introduction to Large Language Models (LLMs); Hugging Face Transformers library — pipelines and tokenisers.	**6**
	Total lectures	**42**

Suggested Text Book(s):

1. Bishop, C. M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Jurafsky, D. and Martin, J. H., Speech and Language Processing, 3rd Edition (Draft), Pearson, 2023. [Available free online]
3. Géron, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2022.
4. Goldberg, Y., Neural Network Methods for Natural Language Processing, Morgan & Claypool Publishers, 2017.

Suggested Reference Book(s):

1. Mitchell, T. M., Machine Learning, McGraw-Hill Education, 1997.
2. Manning, C. D. and Schütze, H., Foundations of Statistical Natural Language Processing, MIT Press, 1999.
3. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
4. Tunstall, L., von Werra, L., and Wolf, T., Natural Language Processing with Transformers, O'Reilly Media, 2022.

Other useful resource(s):

1. Hugging Face Transformers Documentation: https://huggingface.co/docs/transformers
2. spaCy Documentation and Tutorials: https://spacy.io
3. NLTK Book (Free): https://www.nltk.org/book
4. fast.ai Practical Deep Learning for Coders: https://course.fast.ai
5. NPTEL – Introduction to Machine Learning (IIT Madras): https://nptel.ac.in/courses/106106139
6. NPTEL – Deep Learning for NLP: https://nptel.ac.in/courses/106106198
7. Papers With Code – NLP Benchmarks: https://paperswithcode.com/area/natural-language-processing
8. Stanford CS224N – NLP with Deep Learning (Lecture Notes & Videos): https://web.stanford.edu/class/cs224n

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	1.6
CO-2	3	2	2	2	2	1	1	1	2	1	2	1.7
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-5	2	3	2	3	2	1	1	1	2	2	3	2.0
Average	**2.8**	**2.6**	**2.4**	**2.0**	**2.4**	**1.0**	**1.0**	**1.0**	**2.0**	**1.6**	**2.2**	

Ethical Hacking & Prevention

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of Computer Networks and Operating Systems

Course Objectives:

1. To understand the fundamentals of ethical hacking, cybersecurity, and threat landscape.
2. To learn various attack techniques used by hackers and their prevention mechanisms.
3. To analyze vulnerabilities in systems, networks, and web applications.
4. To apply penetration testing tools and methodologies in real-world scenarios.
5. To develop secure systems by implementing best security practices and counter measures.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain fundamentals of ethical hacking, cybersecurity concepts, and types of threats.	Understand
CO-2	Apply various scanning, enumeration, and vulnerability assessment techniques.	Apply
CO-3	Analyze different attack vectors including network, system, and web-based attacks.	Analyze
CO-4	Evaluate security tools and perform penetration testing.	Evaluate
CO-5	Design secure systems and implement preventive measures against cyber threats.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Ethical Hacking: Basics of Cybersecurity, Types of Hackers, Cyber Laws, Footprinting	4
2	Footprinting& Reconnaissance: WHOIS, DNS, Social Engineering	5
3	Scanning & Enumeration: Port Scanning, Vulnerability Scanning, SNMP, LDAP	6
4	System Hacking: Password Cracking, Malware, Privilege Escalation	6
5	Network Attacks & Prevention: DoS/DDoS, MITM, Firewalls, IDS/IPS	7
6	Web Application Security: SQL Injection, XSS, CSRF	7
7	Wireless & Mobile Security: Wi-Fi Attacks, Mobile Security	4
8	Penetration Testing & Countermeasures: Tools and Phases	3

Total lectures	42

Suggested Text Book(s):

1. Kevin Mitnick, The Art of Invisibility, Little, Brown and Company, 2017
2. Michael T. Simpson, Ethical Hacking and Network Defense, Cengage Learning, 3rd Edition
3. Rafay Baloch, Ethical Hacking and Penetration Testing Guide, CRC Press, 2014
4. William Stallings, Network Security Essentials, Pearson, 6th Edition

Suggested Reference Book(s):

1. Jon Erickson, Hacking: The Art of Exploitation, No Starch Press
2. Georgia Weidman, Penetration Testing: A Hands-On Introduction to Hacking, No Starch Press

Other useful resource(s):

1. OWASP Foundation: https://owasp.org
2. Kali Linux Documentation: https://www.kali.org

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Introduction to Large Scale Database Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Database Management Systems, Data Structures, Operating Systems

Course Objectives:

1. To understand the design challenges of large-scale and distributed database systems.
2. To study modern data storage architectures used in cloud and big data platforms.
3. To analyze scalability, consistency, availability, and fault tolerance in distributed databases.
4. To explore NoSQL databases and large-scale data processing frameworks.
5. To design scalable data management solutions for real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain architecture and principles of large-scale database systems.	Understand
CO-2	Apply distributed database concepts for scalable data storage and processing.	Apply
CO-3	Analyze consistency models, replication strategies, and fault tolerance mechanisms.	Analyze
CO-4	Evaluate NoSQL and big data technologies for different application scenarios.	Evaluate
CO-5	Design scalable database solutions for large-scale real-world systems.	Create

Course Contents:

Unit	Contents	Lectures Required
1	**Introduction to Large-Scale Data Systems:** Evolution from traditional DBMS to distributed databases, data-intensive applications, scalability challenges, and centralized vs distributed databases, overview of cloud data platforms.	5
2	**Distributed Database Architecture:** Data partitioning (sharding), replication strategies, distributed query processing, distributed transactions, two-phase commit protocol, distributed indexing.	7
3	**Consistency and CAP Theorem:** Consistency models, CAP theorem, ACID vs BASE properties, eventual consistency, quorum-based systems, concurrency control in distributed environments.	6
4	**NoSQL Database Systems:** Key-value stores, document databases, column-family databases, graph databases; systems such as MongoDB, Cassandra, HBase; schema design for scalability.	7

5	**Large-Scale Data Processing Frameworks:** MapReduce paradigm, Hadoop ecosystem, Spark architecture, distributed storage using HDFS, batch vs stream processing concepts.	6
6	**Fault Tolerance and Reliability:** Data replication, failure detection, recovery mechanisms, consensus algorithms (Paxos/Raft overview), load balancing, distributed monitoring.	6
7	**Cloud Databases and Modern Data Platforms:** Database-as-a-Service (DBaaS), serverless databases, data lakes and data warehouses, real-time analytics systems, security and privacy in large-scale databases.	5
	Total lectures	**42**

Suggested Text Book(s)

1. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media.
2. Hector Garcia-Molina, Jeffrey Ullman & Jennifer Widom, Database Systems: The Complete Book, Pearson.
3. Raghu Ramakrishnan & Johannes Gehrke, Database Management Systems, McGraw-Hill.Tom White, Hadoop: The Definitive Guide, O'Reilly.

Suggested Reference Book(s)

1. Ozsu&Valduriez, Principles of Distributed Database Systems, Springer.
2. George Coulouris et al., Distributed Systems: Concepts and Design, Pearson.
3. Bill Chambers &MateiZaharia, Spark: The Definitive Guide, O'Reilly.
4. Kleppmann Lecture Notes on Distributed Systems (University of Cambridge).

Other useful resource(s):

1. MIT OpenCourseWare – Distributed Computer Systems Engineering (6.824): https://ocw.mit.edu/courses/6-824-distributed-computer-systems-engineering-spring-2006/
2. NPTEL – Big Data Computing (IIT Patna): https://onlinecourses.nptel.ac.in/noc24_cs130/
3. NPTEL – Distributed Systems: https://onlinecourses-archive.nptel.ac.in/noc17_cs42/
4. Apache Hadoop Official Documentation: https://hadoop.apache.org/docs/
5. MongoDB University Courses: https://university.mongodb.com

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	2	1	1	2	**1.8**
CO-2	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-3	3	3	2	3	3	1	1	2	2	2	2	**2.2**
CO-4	2	3	3	3	3	1	1	2	2	2	3	**2.3**
CO-5	3	2	3	2	3	1	1	3	3	2	2	**2.3**
Average	**2.8**	**2.6**	**2.6**	**2.2**	**2.8**	**1.0**	**1.0**	**2.2**	**2.0**	**1.8**	**2.2**	

Introduction to DevOps

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Software Engineering, Cloud Computing

Course Objectives:

1. To understand DevOps principles, culture, and evolution in software development.
2. To apply DevOps practices like CI/CD, TDD, and automation.
3. To analyze cloud-native architectures and microservices.
4. To evaluate collaboration, workflows, and DevOps tools.
5. To measure DevOps performance using metrics and continuous improvement strategies.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain DevOps concepts, principles, and cultural transformation.	Understand
CO-2	Apply DevOps practices like CI/CD, TDD, and automation tools.	Apply
CO-3	Analyze cloud-native architectures and microservices-based systems.	Analyze
CO-4	Evaluate collaboration, workflows, and DevOps lifecycle processes.	Analyze
CO-5	Measure DevOps performance using metrics and continuous improvement strategies.	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to DevOps:** Evolution of software development (Waterfall vs Agile), innovation and business disruption, role of technology, DevOps definition, misconceptions, DevOps culture, characteristics, history and evolution of DevOps movement	**8**
2	**Thinking DevOps:** Social coding, version control and shared repositories, pair programming, working in small batches, MVP concept, feedback loops, TDD, BDD, cloud-native architecture, microservices, resilience engineering	**10**
3	**Working with DevOps:** Software development as craft vs factory, silos and bottlenecks, collaboration, Infrastructure as Code (IaC), ephemeral infrastructure, containers and Docker basics, immutable infrastructure, CI/CD pipelines, automation	**10**
4	**Organizing for DevOps:** Organizational impact, DevOps team structures, cross-functional teams, shared responsibility, misconceptions, consequences in workflows, building shared mindset	**7**

5	**Measuring DevOps:** DevOps metrics (deployment frequency, lead time), social metrics, continuous improvement, actionable metrics, blameless culture, measuring culture, reward systems	**7**
Total lectures		**42**

Suggested Text Book(s):

1. The Phoenix Project – Gene Kim, Kevin Behr, George Spafford
2. The DevOps Handbook – Gene Kim, Jez Humble, Patrick Debois
3. Accelerate – Nicole Forsgren, Jez Humble, Gene Kim

Suggested Reference Book(s):

1. Continuous Delivery – Jez Humble and David Farley
2. Site Reliability Engineering – Betsy Beyer et al.

Other useful resource(s):

1. NPTEL DevOps Courses:https://elearn.nptel.ac.in/shop/completed-courses/short-term-programs-completed/cicd-devops-automation-and-devsecops-automation/
2. IBM DevOps Learning Resources:https://developer.ibm.com/devpractices/devops/tutorials/
3. AWS DevOps Tutorials: https://market.tutorialspoint.com/course/aws-devops-engineer-certification-2022/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	2	1	1	1	2	1	2	1.8
CO-2	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-3	3	2	3	3	3	1	1	1	2	2	2	2.2
CO-4	2	3	2	3	2	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	2	3	2	2	2.2
Average	**2.8**	**2.4**	**2.6**	**2.4**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.0**	

Industrial Automation and IoT

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic Electronics, Microprocessors and Microcontrollers, Computer Networks, Programming in C/Python

Course Objectives:

1. To understand the architecture, components, and hierarchy of modern industrial automation systems including sensors, actuators, PLCs, and DCS.
2. To study SCADA systems, HMI design, and industrial communication protocols used in process control and manufacturing environments.
3. To explore the architecture, hardware platforms, and communication protocols of the Internet of Things and Industrial IoT.
4. To analyse edge computing, digital twin concepts, and IIoT data pipelines for industrial monitoring and predictive maintenance.
5. To examine cybersecurity threats, safety standards, and regulatory frameworks governing industrial automation and IIoT deployments.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of industrial automation, control systems, sensors, actuators, and programmable logic controllers (PLCs) used in modern manufacturing environments.	Understand
CO-2	Explain the architecture, communication protocols, and standards of the Industrial Internet of Things (IIoT) including MQTT, OPC-UA, Modbus, and PROFINET.	Understand
CO-3	Apply SCADA systems, HMI design principles, and industrial networking concepts to model real-world automated process control scenarios.	Apply
CO-4	Apply IoT sensing, edge computing, and data acquisition techniques to design IIoT data pipelines for industrial monitoring and predictive maintenance.	Apply
CO-5	Analyse industrial automation systems and IIoT deployments with respect to reliability, cybersecurity threats, real-time constraints, and safety standards.	Analyse

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Industrial Automation: Evolution of automation — manual, mechanised, automated, and smart manufacturing; automation hierarchy — field level, control level, supervisory level, enterprise level; types of automation — fixed, programmable, flexible; key components — sensors, actuators, controllers, HMI; introduction to Industry 4.0 and cyber-physical systems (CPS); overview of smart factories.	**5**

2	Sensors, Actuators and Signal Conditioning: Classification of sensors — temperature (RTD, thermocouple), pressure, flow, proximity, vision; signal types — analog vs. digital, 4–20 mA current loops; signal conditioning — amplification, filtering, ADC/DAC; actuators — electric motors (DC, stepper, servo), pneumatic and hydraulic actuators, solenoid valves; transducer characteristics — accuracy, resolution, linearity, repeatability.	**6**
3	Programmable Logic Controllers (PLCs) and Control Systems: PLC architecture — CPU, memory, I/O modules, power supply; PLC programming languages (IEC 61131-3) — Ladder Diagram, Function Block Diagram, Structured Text, Sequential Function Chart; closed-loop control — PID controllers, tuning methods; motion control basics; safety PLCs and functional safety standards (IEC 61508, IEC 62061); introduction to Distributed Control Systems (DCS).	**8**
4	SCADA, HMI and Industrial Networking: SCADA system architecture — field devices, RTUs/PLCs, communication network, SCADA server, historian; HMI design principles — alarm management, data visualisation, operator interface; industrial communication protocols — Modbus RTU/TCP, PROFIBUS, PROFINET, EtherNet/IP, OPC and OPC-UA; fieldbus vs. industrial Ethernet; network topologies in automation; introduction to time-sensitive networking (TSN).	**7**
5	Internet of Things (IoT) — Architecture and Protocols: IoT reference architecture — perception, network, and application layers; IoT hardware — microcontrollers (Arduino, ESP32), single-board computers (Raspberry Pi); wireless communication technologies — Wi-Fi, Bluetooth, Zigbee, Z-Wave, LoRa, NB-IoT, 5G; IoT application protocols — MQTT, CoAP, AMQP, HTTP/REST; IoT platforms — AWS IoT, Azure IoT Hub, Google Cloud IoT, Things Board.	**6**
6	Industrial IoT (IIoT) and Edge Computing: IIoT vs. consumer IoT; IIoT reference architecture (RAMI 4.0, IIRA); edge computing in industrial settings — edge gateways, fog nodes; real-time data acquisition and preprocessing; time-series data handling; digital twins — concept, architecture, and industrial applications; OPC-UA over MQTT (Sparkplug B); IIoT use cases — predictive maintenance, condition monitoring, energy management, asset tracking.	**7**
7	Cybersecurity, Safety and Standards in Industrial Automation and IIoT: Cybersecurity threats in OT/ICS environments — ransomware, man-in-the-middle, denial of service; ICS security frameworks — IEC 62443, NIST SP 800-82; network segmentation — DMZ, Purdue model, zero-trust architecture; secure remote access; safety instrumented systems (SIS) and safety integrity levels (SIL); regulatory and interoperability standards — ISO 9001, ISO 50001, IEEE 2413; emerging trends — AI in automation, autonomous robots, collaborative robots (cobots), 5G in IIoT.	**3**
	Total lectures	**42**

Suggested Text Book(s):

1. Groover, M. P., Automation, Production Systems, and Computer-Integrated Manufacturing, 4th Edition, Pearson, 2015.
2. Mackay, S., Wright, E., Reynders, D., and Park, J., Practical Industrial Data Networks: Design, Installation and Troubleshooting, Newnes, 2004.
3. Bahga, A. and Madisetti, V., Internet of Things: A Hands-On Approach, VPT, 2014.
4. Bolton, W., Programmable Logic Controllers, 6th Edition, Newnes, 2015.

Suggested Reference Book(s):

1. Mahalik, N. P., Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control, Springer, 2003.

2. Buyya, R. and Dastjerdi, A. V., Internet of Things: Principles and Paradigms, Morgan Kaufmann, 2016.
3. Zurawski, R. (Ed.), Industrial Communication Technology Handbook, 2nd Edition, CRC Press, 2014.
4. Benaben, F. and Stuber, W. (Eds.), Interoperability in Digital Industry, Springer, 2021.

Other useful resource(s):

1. NPTEL – Industrial Automation (IIT Kharagpur): https://nptel.ac.in/courses/10710506.
2. NPTEL – Internet of Things (IIT Bombay): https://nptel.ac.in/courses/108101093
3. Siemens Automation Learning Portal: https://support.industry.siemens.com
4. Eclipse IoT (Open-source IoT Frameworks): https://iot.eclipse.org
5. ThingsBoard IoT Platform (Open-source): https://thingsboard.io
6. IEC 61131-3 PLC Programming Reference: https://plcopen.org
7. Papers With Code – IoT and Edge Computing: https://paperswithcode.com

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	2	1	2	1	2	1.7
CO-2	3	2	2	2	3	1	1	1	2	1	2	1.8
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	3	3	2	2	3	1	1	1	2	2	2	2.0
CO-5	2	3	2	3	2	2	2	2	2	2	2	2.2
Average	**2.8**	**2.6**	**2.2**	**2.0**	**2.6**	**1.2**	**1.4**	**1.2**	**2.0**	**1.6**	**2.0**	

Explainable AI

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning and Deep Learning

Course Objectives:

1. To introduce the fundamentals of Explainable Artificial Intelligence (XAI) and its significance in machine learning models.
2. To understand interpretability concepts and techniques for explaining black-box models.
3. To explore deep learning explanation methods such as attention mechanisms, feature visualization, and Grad-CAM.
4. To study different XAI models including ante-hoc, post-hoc, and hybrid approaches.
5. To apply XAI techniques and evaluate trust, reliability, and performance of AI systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the fundamental concepts of Explainable AI including taxonomy, causality, and interpretability in machine learning.	Understand
CO-2	Apply interpretability techniques to machine learning models such as regression, decision trees, and neural networks.	Apply
CO-3	Analyze deep learning explanation methods including attention mechanisms, feature visualization, Grad-CAM, and sensitivity analysis.	Analyze
CO-4	Evaluate different XAI models such as ante-hoc, post-hoc, and hybrid approaches for explaining black-box systems.	Evaluate
CO-5	Apply XAI techniques such as LIME, SHAP, DiCE, and LRP and assess trust and performance of AI systems using evaluation metrics.	Apply

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Explainable Artificial Intelligence:** Fundamentals of XAI, Categorization of XAI,Taxonomy of XAI methods for Machine Learning , Machine Learning Interpretability, Causal Model Induction, Causality learning, XAI techniques and limitations.	6
2	**Interpretability:** Transaction Difference between Interpretability and Explainability, Interpretability methods to explain Black-Box Model, Scope of Interpretability, Apply interpretability on Regression, Logistic regression, Generalized Additive Models, Decision Tree, Neural network interpretation, Evaluation of Interpretability	7
3	**Deep Explanation:** Attention Mechanisms, Modular Networks, Feature Identification, Learn to Explain - Feature Visualization, Deep	7

	Visualization- gradcam and Activation maps - Sensitivity analysis	
4	**XAI Model:** Ante-hoc Explainability (AHE) models, Post-hoc Explainability (PHE) models, Interactive Machine Learning (IML), Black Box Explanation through Transparent Approximation (BETA) models - Hybrid Models.	**8**
5	**XAI Methods:** XAI Techniques - Local Interpretable Model-Agnostic Explanations (LIME), Understanding Mathematical representation of LIME, Shapley Additive exPlanations (SHAP), Diverse Counterfactual Explanations (DiCE), Layer wise Relevance Propagation (LRP).	**9**
6	**Trust and acceptance:**Metrics to evaluate XAI, Trustworthy Explainability Acceptance, Power Quality Disturbance (PQD) classification, Methods for measuring human intelligence. Evaluating AI system.	**5**
Total lectures		**42**

Suggested Text Book(s):

1. Molnar, Christoph. "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2019. https://christophm.github.io/interpretable-ml-book/.
2. Explainable Artificial Intelligence: An Introduction to Interpretable Machine Learning, Uday Kamath: John Liu, Springer, ISBN 9783030833558

Suggested Reference Book(s):

1. Tim Miller Explanation in Artificial Intelligence: Insight from Social Science, https://arxiv.org/abs/1706.07269
2. A Guide for making black-box machine learning models https://christophm.github.io/interpretable-ml-book
3. Explainable AI: A Review of Machine Learning Interpretability Methods https://www.mdpi.com/1099-4300/23/1/18
4. Lötsch, J.; Kringel, D.; Ultsch, A. Explainable Artificial Intelligence (XAI) in Biomedicine: Making AI Decisions Trustworthy for Physicians and Patients. BioMedInformatics 2022, 2, 1-17. https://doi.org/10.3390/biomedinformatics2010001

Other useful resource(s):

1. Link to NPTEL course contents: Explainable AI - Technical Explanation and Transparency

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	1	1	2	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	1	2	**1.9**
CO-3	3	3	2	3	3	1	1	1	2	1	3	**2.1**
CO-4	3	3	3	3	2	2	2	1	2	1	3	**2.3**
CO-5	3	3	3	3	3	2	2	2	3	2	3	**2.6**
Average	**3**	**2.8**	**2.2**	**2.4**	**2.6**	**1.4**	**1.4**	**1.2**	**2**	**1.2**	**2.6**	

Probabilistic Graphical Models

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine learning and Probability

Course Objectives:

1. Learn introductory concepts in probabilistic graphical models.
2. Able to model problems using graphical models.
3. Able to model problems using design inference algorithms.
4. Learn the structure of the graphical model from data

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand fundamentals of probability, graphs, and probabilistic graphical models.	Understand
CO-2	Analyze and construct Bayesian Networks and Markov Random Fields.	Analyse
CO-3	Apply advanced models like DBNs, HMMs, and Gaussian networks.	Apply
CO-4	Implement exact and approximate inference algorithms.	Apply
CO-5	Develop learning methods including MLE, Bayesian learning, and EM.	Create
CO-6	Apply PGMs to causal reasoning and decision-making problems.	Apply

Course Contents:

Unit	Contents	Lectures required
1	**Foundations:** Introduction to probabilistic models, structured probabilistic models and motivation, basics of probability theory, random variables and joint distributions, independence and conditional independence, graph theory basics including nodes, edges, and paths.	6
2	**Bayesian Networks:** Introduction to Bayesian networks, factorization and conditional probability tables, independence in Bayesian networks, d-separation, I-maps and perfect maps, constructing Bayesian networks from data, case studies such as Naive Bayes and student model.	7
3	**Undirected Graphical Models:** Markov networks, factors and Gibbs distributions, independence in Markov networks, conversion between Bayesian networks and Markov networks, conditional random fields.	5
4	**Advanced Representations:** Local conditional probability distributions, context-specific independence, noisy-OR and generalized linear models, dynamic Bayesian networks and hidden Markov models, Gaussian networks and exponential family.	5

5	**Exact Inference:** Inference problem formulation, variable elimination algorithm, complexity of inference, clique trees and junction trees, message passing using sum-product algorithm.	**5**
6	**Approximate Inference**: Inference as optimization, belief propagation, variational inference using mean field methods, sampling methods including Monte Carlo techniques, Markov chain Monte Carlo and Gibbs sampling.	**5**
7	**MAP and Temporal Inference:** MAP inference using max-product algorithm, graph cuts and optimization methods, inference in temporal models including filtering and particle filters.	**3**
8	**Learning in Probabilistic Graphical Models**: Learning problem and overfitting, parameter estimation using maximum likelihood and Bayesian methods, structure learning using score-based and constraint-based approaches, learning with hidden variables using expectation maximization.	**4**
9	**Advanced Topics:** Causality and causal graphs, decision making and influence diagrams	**2**
Total lectures		**42**

Suggested Text Book(s):

1. Koller,D.andFriedman,N.(2009).ProbabilisticGraphicalModels:PrinciplesandTechniques.MIT Press

Suggested Reference Book(s):

1. Jensen, F. V. and Nielsen, T. D. (2002). Bayesian Networks and Decision Graphs. Information Science and Statistics. Springer, 2ndedition.
2. KevinP.Murphy(2013)MachineLearning:AProbabilisticPerspective.4thPrinting.MITPress.
3. Barber,D.(2011).BayesianReasoningandMachineLearning.CambridgeUniversityPress,1stedition.
4. Bishop, C. M. (2011). Pattern Recognition and Machine Learning(Information Science and Statistics). Springer, 2nd printing.
5. Wainwright,M.andJordan,M.(2008).GraphicalModels,ExponentialFamilies,andVariational Inference. Foundations and Trends in Machine Learning,1:1–305.

Other useful resource(s):

1. Conference papers and journalarticles
2. Probabilistic Graphical Models Stanford Course — best structured course with theory + applications.
3. CS228: Probabilistic Graphical Models — lecture videos + notes.
4. MIT 6.438 Algorithms for Inference — strong on inference algorithms.

Evaluation Scheme:

S. No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2										2.3
CO-2	3	3	1	2								2.2
CO-3	3	2	2	2	3							2.3
CO-4	2	3	1	2	2	2	1	2				1.8
CO-5	3	3	2	2	3		1					2.3
CO-6	3	3	3	3	3	2	2	1	2	2	2	2.4
Average	2.8	2.6	1.8	2.2	2.8	2	1.3	1.5	2	2	2	

AI for IoT

COURSE CODE: XXXX
COURSE CREDITS: 3
CORE/ELECTIVE: ELECTIVE
L-T-P: 3-0-0

Pre-requisite: Programming in Python, Introduction to Machine Learning, Basics of Computer Networks

Course Objectives:

1. To understand the convergence of Artificial Intelligence with Internet of Things — architectures, communication, and data ecosystems.
2. To apply machine learning techniques for processing, analysing, and extracting insights from IoT sensor data streams.
3. To implement deep learning models including CNNs, LSTMs, and Transformers for time-series and image-based IoT tasks.
4. To design and deploy Edge AI and Tiny ML solutions on resource-constrained IoT devices with model optimization strategies.
5. To evaluate AI-driven IoT applications in smart healthcare, industrial IoT, smart cities, and autonomous systems, with attention to security and privacy.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the architecture of IoT systems and articulate the role of AI in enabling intelligent, data-driven IoT applications.	Understand
CO-2	Apply machine learning algorithms to preprocess, analyse, and extract insights from IoT-generated time-series and sensor data.	Apply
CO-3	Implement deep learning models — CNNs, LSTMs, and Transformer-based architectures — for IoT data classification and prediction tasks.	Apply
CO-4	Design, optimise, and deploy TinyML and Edge AI models on microcontrollers and edge devices for real-time, low-latency inference.	Create
CO-5	Evaluate AI-driven IoT systems for performance, security, and privacy, and propose solutions for real-world deployment challenges.	Evaluate

Course Contents:

Unit	Contents	Lectures Required
1	Introduction to AI and IoT Convergence: Overview of the IoT ecosystem — sensing, connectivity, processing, and actuation layers; Role of AI in IoT — from rule-based automation to intelligent, adaptive systems; Characteristics of IoT data: volume, velocity, variety, veracity, and value; Survey of AI-IoT applications — smart homes, precision agriculture, smart	**4**

	healthcare, industrial IoT (IIoT), and autonomous vehicles; IoT reference architectures: three-layer and five-layer models; Comparison of cloud computing, fog computing, and edge computing paradigms for AI deployment.	
2	IoT Communication Protocols and Data Pipelines: Device connectivity standards — MQTT, CoAP, AMQP, HTTP/REST, and WebSockets; Wireless technologies — Zigbee, Z-Wave, LoRaWAN, NB-IoT, and 5G for IoT; IoT cloud platforms — AWS IoT Core, Azure IoT Hub, Google Cloud IoT, and ThingsBoard; Real-time data ingestion with Apache Kafka and Node-RED; Time-series databases — InfluxDB, TimescaleDB, and OpenTSDB; Data lake architectures for scalable IoT data storage and retrieval.	**5**
3	IoT Data Preprocessing and Feature Engineering: Sensor data quality issues — missing values, noise, drift, and outliers; Signal smoothing techniques — moving average, Kalman filtering, and Savitzky-Golay filter; Sliding window and tumbling window segmentation for time-series; Feature extraction from time-domain (mean, variance, RMS, skewness, kurtosis) and frequency-domain (FFT, PSD) signals; Wavelet transform for multi-resolution signal analysis; Normalisation, standardisation, and dimensionality reduction using PCA and autoencoders.	**5**
4	Machine Learning for IoT: Supervised learning — regression and classification on sensor data using Decision Trees, Random Forests, and Gradient Boosting; Unsupervised learning — k-means, DBSCAN for IoT device clustering; Anomaly detection using Isolation Forest, Local Outlier Factor (LOF), and One-Class SVM; AutoEncoder-based anomaly detection for sensor streams; Predictive maintenance — remaining useful life (RUL) estimation; Federated Learning for privacy-preserving collaborative model training across IoT devices; AutoML tools for IoT: Google AutoML, Auto-sklearn.	**6**
5	Deep Learning for IoT: Convolutional Neural Networks (CNNs) for image-based and spectro-temporal IoT tasks; Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) for time-series forecasting; Temporal Convolutional Networks (TCNs) and WaveNet for sequence modelling; Transformer and Time-Series Transformer (PatchTST, Informer) for long-range IoT sequence modelling; Hybrid CNN-LSTM architectures for activity recognition and fault detection; Generative AI for synthetic IoT data augmentation — VAE and GAN-based approaches; Transfer learning and domain adaptation for IoT tasks with limited labelled data.	**6**
6	Edge AI and TinyML: Edge computing architecture — microcontrollers, edge servers, and gateway nodes; TinyML fundamentals — deploying trained models on ultra-low-power devices (Arduino Nano 33, STM32, ESP32); Model optimisation techniques: post-training quantisation (INT8/FP16), weight pruning, and knowledge distillation; Neural Architecture Search (NAS) for resource-constrained hardware; Deployment frameworks — TensorFlowLite Micro, Edge Impulse, ONNX Runtime, and OpenVINO; On-device learning and continual learning for adaptive IoT systems; Benchmarking edge AI: latency, throughput, power consumption,	**7**

	and memory footprint trade-offs; Case study: end-to-end TinyML pipeline for keyword spotting and gesture recognition.	
7	AI for IoT Security and Privacy: IoT threat landscape — denial-of-service, spoofing, replay attacks, and firmware exploits; AI-based network intrusion detection systems (NIDS) forIoT; Deep learning for malware classification in IoT firmware; Adversarial attacks on IoT AI models — evasion, poisoning, and model inversion; Defences — adversarial training, certified robustness, and anomaly-based filtering; Differential privacy and secure multi-party computation in federated IoT; Blockchain-based trust and identity management for IoT ecosystems.	**5**
8	Real-World Applications and Emerging Trends: Smart healthcare — remote patient monitoring, fall detection, and wearable AI; Smart cities — traffic optimisation, energy management, and environmental sensing; Industrial IoT — predictive maintenance, quality control, and digital twins; AI-powered agriculture — crop monitoring, irrigation automation, and yield prediction; Large Language Models (LLMs) and Generative AI for IoT data summarisation and natural language interfaces; AIoT deployment challenges: scalability, interoperability, regulatory compliance, and ethical considerations.	**4**
Total lectures		**42**

Suggested Text Book(s):

1. Vijay Madisetti, ArshdeepBahga, Internet of Things: A Hands-On Approach, Universities Press, 2nd Edition, 2021.
2. Pete Warden, Daniel Situnayake, TinyML: Machine Learning with TensorFlowLite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media, 2020.
3. Gian Marco Iodice, AI at the Edge: Solving Real-World Problems with Embedded Machine Learning, O'Reilly Media, 2023.
4. AurélienGéron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.

Suggested Reference Book(s):

1. Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds.), Architecting the Internet of Things, Springer, 2011.
2. Francois Chollet, Deep Learning with Python, Manning Publications, 2nd Edition, 2021.
3. Shiv Shankar Shukla, Internet of Things with AI, BPB Publications, 2022.
4. Marco Schwartz, Building Smart Homes with Raspberry Pi Zero, Packt Publishing, 2016.

Other useful resource(s):

1. NPTEL course — Internet of Things: https://nptel.ac.in/courses/106105166
2. TensorFlowLite for Microcontrollers: https://www.tensorflow.org/lite/microcontrollers
3. Edge Impulse documentation and tutorials: https://docs.edgeimpulse.com
4. AWS IoT Core getting started guide: https://aws.amazon.com/iot-core/getting-started/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	3	2	3	1	1	2	2	2	2	**2.1**
CO-4	3	3	3	3	3	1	1	2	2	2	2	**2.3**
CO-5	3	3	3	2	3	2	2	2	3	2	2	**2.5**
Average	**3.0**	**2.6**	**2.8**	**2.0**	**2.8**	**1.2**	**1.2**	**1.6**	**2.2**	**1.8**	**2.0**	**2.1**

Sentiment Analysis and Opinion Mining

COURSE CODE: XXXX
COURSE CREDITS: 3
CORE/ELECTIVE: ELECTIVE
L-T-P: 3-0-0

Pre-requisite: Natural Language Processing, Machine Learning, Python Programming

Course Objectives:

1. To understand the theoretical foundations of sentiment analysis, opinion mining, and subjectivity analysis at document, sentence, and aspect levels.
2. To apply lexicon-based methods and classical machine learning techniques for polarity detection and opinion classification.
3. To implement deep learning architectures — CNNs, LSTMs, and pre-trained transformer models (BERT, RoBERTa) — for sentiment analysis tasks.
4. To analyse aspect-level opinions, fine-grained sentiment, and emotion detection from text and multimodal data.
5. To design and evaluate end-to-end sentiment analysis pipelines for real-world applications including social media analytics and market intelligence.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the definitions of opinion, sentiment, subjectivity, and polarity and differentiate the levels and tasks of sentiment analysis.	Understand
CO-2	Apply lexicon-based and classical machine learning methods for document-level and sentence-level sentiment classification.	Apply
CO-3	Implement deep learning models — LSTMs, CNNs, and fine-tuned transformers (BERT, RoBERTa) — for sentiment and opinion mining tasks.	Apply
CO-4	Analyse aspect-based sentiment, comparative opinion, emotion detection, and sarcasm from text and multimodal data sources.	Analyse
CO-5	Design, evaluate, and interpret complete sentiment analysis pipelines for real-world business and social intelligence applications.	Create

Course Contents:

Unit	Contents	Lectures Required
1	Introduction to Sentiment Analysis and Opinion Mining: Definitions — opinion, sentiment, subjectivity, polarity, and stance; Levels of sentiment analysis — document, sentence, aspect, and entity level; Opinion components — opinion holder, opinion target, opinion expression, and	**4**

	time; Subjectivity detection vs. polarity classification; Challenges — irony, sarcasm, domain dependence, negation, and linguistic ambiguity; Applications — product reviews, social media monitoring, financial sentiment, healthcare feedback, and political opinion analysis.	
2	Text Preprocessing and Representation for Sentiment Analysis: Tokenisation, sentence segmentation, stopword removal, lemmatisation, and stemming; Handling social media text — slang, abbreviations, hashtags, emojis, and code-switching; Bag-of-Words (BoW), TF-IDF, and n-gram feature representations; Static word embeddings — Word2Vec (Skip-gram, CBOW), GloVe, and FastText; Handling negation and degree modifiers in feature engineering; Contextual embeddings — ELMo, and introduction to transformer-based representations.	**4**
3	Lexicon-based Approaches for Sentiment Analysis: Sentiment lexicons — SentiWordNet, VADER, AFINN, NRC Emotion Lexicon, and Opinion Lexicon (Bing Liu); Dictionary-based and corpus-based methods for lexicon construction and expansion; Semantic orientation using PMI and LSA; Rule-based sentiment scoring with modifiers, negations, and intensifiers; Handling of discourse structure in lexicon-based models; Comparative analysis of lexicons across domains; Advantages, limitations, and domain adaptation of lexicon-based methods.	**5**
4	Machine Learning for Sentiment Analysis: Feature engineering — BoW, TF-IDF, POS tags, dependency relations, and sentiment scores; Classical classifiers — Naive Bayes, Logistic Regression, SVM, and Random Forest for sentiment classification; Ensemble methods — bagging, boosting, and stacking for sentiment; Cross-domain and cross-lingual sentiment classification; Semi-supervised and weakly supervised sentiment learning; Evaluation metrics — accuracy, macro/micro F1-score, precision, recall, and AUC-ROC; Error analysis and model interpretability with LIME and SHAP.	**5**
5	Deep Learning for Sentiment Analysis: CNN-based models for sentence-level sentiment — Kim's TextCNN architecture; LSTM and Bidirectional LSTM for sequential sentiment modelling; Attention mechanisms — self-attention and hierarchical attention networks; Hierarchical Attention Network (HAN) for document-level classification; Graph Convolutional Networks (GCN) for syntax-aware sentiment analysis; Training strategies — dropout, batch normalisation, and regularisation for NLP models; Benchmark datasets — SST, IMDb, Yelp, Amazon Reviews.	**6**
6	Transformer-based Models for Sentiment Analysis: Transformer architecture — self-attention, positional encoding, encoder-decoder framework; BERT pre-training objectives — Masked Language Modelling (MLM) and Next Sentence Prediction (NSP); Fine-tuning BERT for sentiment classification — sentence-pair and single-sentence tasks; Domain-specific BERT variants — FinBERT, BioBERT, TweetBERT; RoBERTa, ALBERT, and DistilBERT for efficient sentiment modelling; Prompt-based learning and in-context learning with GPT-series LLMs for zero-shot and few-shot sentiment; Multilingual sentiment analysis with mBERT and XLM-RoBERTa.	**6**

7	Aspect-Based Sentiment Analysis (ABSA): Tasks in ABSA — aspect term extraction (ATE), aspect category detection (ACD), and aspect-level polarity classification (ALPC); Pipeline vs. end-to-end joint ABSA models; Transformer-based ABSA — BERT-pair, GRACE, InstructABSA, and Generative ABSA (PARAPHRASE); Implicit aspect and opinion extraction; Comparative opinion mining — identification of comparison relations and preferred entities; Argument mining — claim and premise detection in opinionated text; ABSA benchmark datasets — SemEval 2014/2015/2016 ABSA tasks.	**6**
8	Multimodal Sentiment Analysis, Emotion Detection, and Advanced Topics: Multimodal sentiment from text, audio, and video — feature extraction and fusion strategies; Early, late, and cross-modal attention fusion; Emotion detection — basic emotions (Ekman) vs. dimensional models (valence-arousal-dominance); Fine-grained sentiment and intensity estimation; Sarcasm and irony detection — incongruity-based and context-aware models; Hate speech, offensive language, and stance detection; Datasets — CMU-MOSI, CMU-MOSEI, SemEval, TweetEval; Ethical considerations — bias in sentiment models and fairness-aware training.	**6**
Total lectures		**42**

Suggested Text Book(s):

1. Bing Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2nd Edition, 2020.
2. Bo Pang, Lillian Lee, Opinion Mining and Sentiment Analysis, Foundations and Trends in Information Retrieval, Now Publishers, 2008.
3. Yoav Goldberg, Neural Network Methods for Natural Language Processing, Morgan & Claypool Publishers, 2017.
4. Lewis Tunstall, Leandro von Werra, Thomas Wolf, Natural Language Processing with Transformers, O'Reilly Media, Revised Edition, 2022.

Suggested Reference Book(s):

1. Christopher Manning, HinrichSchütze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
2. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, MIT Press, 2016.
3. AurélienGéron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 3rd Edition, 2022.
4. Steven Bird, Ewan Klein, Edward Loper, Natural Language Processing with Python, O'Reilly Media, 2009.

Other useful resource(s):

1. NPTEL course — Natural Language Processing: https://nptel.ac.in/courses/106105158
2. Hugging Face Transformers documentation: https://huggingface.co/docs/transformers
3. SemEval shared tasks on Sentiment Analysis: https://semeval.github.io
4. NLTK and spaCy documentation: https://www.nltk.org and https://spacy.io/usage

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	2	1	1	1	2	1	2	**1.7**
CO-2	3	3	2	2	3	1	1	1	2	2	2	**2.0**
CO-3	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-4	3	3	3	3	3	1	1	2	3	2	2	**2.4**
CO-5	3	3	3	2	3	2	2	2	3	2	2	**2.5**
Average	**3.0**	**2.8**	**2.6**	**2.2**	**2.8**	**1.2**	**1.2**	**1.4**	**2.4**	**1.8**	**2.0**	**2.1**

AI for Healthcare & Smart Systems

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning, Data Structures, Probability & Statistics, Python Programming

Course Objectives:

1. To understand the fundamentals of Artificial Intelligence in healthcare and smart systems.
2. To explore machine learning and deep learning techniques for medical data analysis.
3. To design intelligent healthcare solutions such as disease prediction and diagnosis systems.
4. To analyze smart systems including IoT-based healthcare monitoring and automation.
5. To address ethical, privacy, and security challenges in AI-based healthcare systems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain AI concepts, healthcare data types, and smart system architectures.	Understand
CO-2	Apply machine learning and deep learning techniques to healthcare datasets.	Apply
CO-3	Design AI-based models for disease prediction, diagnosis, and monitoring.	Apply
CO-4	Analyze challenges in healthcare AI such as data privacy, bias, and system reliability.	Analyze
CO-5	Develop smart healthcare systems using AI, IoT, and real-time data processing.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to AI in Healthcare & Smart Systems**: Overview of AI, Applications in Healthcare, Smart Healthcare Systems, Types of Healthcare Data (EHR, Imaging, Sensor Data), Challenges in Healthcare AI	1
2	**Machine Learning for Healthcare**: Supervised & Unsupervised Learning, Classification & Regression in Healthcare, Feature Engineering, Model Evaluation (Accuracy, Precision, Recall, ROC), Case Studies (Disease Prediction)	8
3	**Deep Learning in Healthcare**: Neural Networks, CNN for Medical Imaging, RNN/LSTM for Time-Series Health Data, Transfer Learning, Applications in MRI, X-ray, and ECG analysis	3
4	**Smart Healthcare Systems & IoT**: IoT Architecture, Wearable Devices, Remote Patient Monitoring, Edge Computing, Real-time Data Processing, Smart Hospitals	3

5	**AI in Diagnosis & Decision Support Systems**: Clinical Decision Support Systems (CDSS), Natural Language Processing for Medical Records, Chatbots in Healthcare, Personalized Medicine	**6**
6	**Data Privacy, Security & Ethics**: HIPAA basics, Data Privacy Issues, Bias in AI Models, Ethical Considerations, Explainable AI (XAI) in Healthcare	**9**
7	**Deployment of AI Models in Healthcare**: Model Deployment, Cloud-based Healthcare Systems, APIs for AI Models, Integration with Healthcare Systems ,Smart Cities Healthcare Integration, Pandemic Prediction Models, AI in Drug Discovery, Future Trends in AI Healthcare	**12**
Total lectures		**42**

Suggested Text Book(s):

1. Kevin Murphy, *Machine Learning: A Probabilistic Perspective*.
2. Gyorgy J. Simon, Constantin Aliferis, *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences*
3. Rajkomar et al., *Machine Learning for Healthcare Applications*.

Suggested Reference Book(s):

1. Tom Mitchell, *Machine Learning*
2. Ian Goodfellow, *Deep Learning*, MIT Press
3. Healthcare AI research papers (IEEE, Springer, Elsevier)

Other useful resource(s):

1. Udemy https://www.udemy.com/course/artificial-intelligence-in-health-care/
2. Coursera: AI for Medicine Specialization https://www.coursera.org/specializations/ai-healthcare
3. WHO Digital Health Resources https://www.who.int/health-topics/digital-health#tab=tab_1

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	2	1	2	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	2	2	**2**
CO-3	3	2	3	3	3	2	1	1	2	2	2	**2.1**
CO-4	2	3	2	3	2	3	3	1	2	2	3	**2.3**
CO-5	3	2	3	2	3	2	2	2	3	2	3	**2.4**
Average	**2.8**	**2.4**	**2.2**	**2.2**	**2.6**	**1.8**	**1.6**	**1.2**	**2.2**	**1.8**	**2.4**	

Digital Forensics and Cyber Laws

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Cryptography &Security, Computer Networks

Course Objectives:

1. To understand the legal framework governing cyberspace and cyber activities.
2. To explore practical knowledge about ethical hacking methodology within legal boundaries.
3. To learn the importance of legal procedures in evidence handling and storage for various devices.
4. To develop an understanding of current cybersecurity issues and applicable cyber laws.
5. To investigate cyber incidents, IDS, technical exploits, and network-related legal implications.
6. To apply digital forensic knowledge using forensic tools and investigation report writing in compliance with legal standards.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Introduction to Cyber Laws and Ethical Hacking	Understand
CO-2	Introduction to Digital Forensics and Digital Evidences	Understand
CO-3	Computer Security Incident Response Methodology	Apply
CO-4	Forensic Duplication, Disk Analysis, and Investigation Data Analysis	Analyze
CO-5	Network Forensics Incidents, Using Routers as Response Tools	Apply
CO-6	Forensic Investigation Report and Forensic Tools	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Cyber Laws:** Overview of Cyber Laws, Evolution of Cyber Laws, Legal Framework in Cyberspace, Jurisdiction Issues in Cyber Law, Cyber Law in India (IT Act 2000 and Amendments), Digital Signatures and Electronic Records, Data Protection and Privacy Laws, Intellectual Property Rights in Cyberspace. **Ethical Hacking:** Difference between Hacking and Ethical Hacking, Legal Boundaries of Ethical Hacking, Steps of Ethical Hacking, Exploring tools for ethical hacking: reconnaissance tools, scanning tools.	**7**

2	Digital Forensic, Rules for Digital Forensic The Need for Digital Forensics, Types of Digital Forensics, Ethics in Digital Forensics, Digital Evidences: Types and characteristics and challenges for Evidence Handling.	**7**
3	Introduction to Computer Security Incident Goals of Incident Response, Incident Response Methodology, Formulating Response Strategy, IR Process – Initial Response, Investigation, Remediation, Tracking of Significant, Investigative Information, Reporting Pre Incident Preparation, Incident Detection, and Characterization. Live Data Collection: Live Data Collection on Microsoft Windows Systems: Live Data Collection on Unix-based Systems	**7**
4	Forensic Duplication Forensic Image Formats, Traditional Duplication, Live System Duplication, Forensic Duplication tools Disk and File System Analysis: Media Analysis Concepts, File System Abstraction Model The Sleuth Kit: Installing the Sleuth Kit, Sleuth Kit Tools Partitioning and Disk Layouts: Partition Identification and Recovery, Redundant Array of Inexpensive Disks Special Containers: Virtual Machine Disk Images, Forensic Containers Hashing, Carving: Foremost, Forensic Imaging: Deleted Data, File Slack, Data Analysis, Analysis Methodology Investigating Windows systems, Investigating UNIX systems, Investigating Applications, Web Browsers, Email, Malware Handling: Static and Dynamic Analysis	**10**
5	Technical Exploits and Password Cracking (Legal Implications), Introduction to Intrusion Detection Systems (IDS), Types of IDS Understanding Network Intrusions and Attacks, Analyzing Network Traffic, Collecting Network-based Evidence, Legal Aspects of Evidence Handling Investigating Routers, Handling Router Table Manipulation Incidents, Using Routers as Response Tools	**6**
6	Forensic Reporting: Goals of Report, Layout of Investigative Report, Legal Guidelines for Writing Reports, Chain of Custody, Admissibility of Digital Evidence in Court, Sample Forensic Reports Computer Forensic Tools: Need and Types of Tools, Tasks Performed Study of Open-source Tools like SFIT, Autopsy for acquiring, searching, analyzing, and storing digital evidence	**5**
	Total lectures	**42**

Suggested Text Book(s):

1. Jason Luttgens, Matthew Pepe, Kevin Mandia, *"Incident Response and Computer Forensics"*, 3rd Edition, Tata McGraw Hill, 2014.
2. Nilakshi Jain, Dhananjay Kalbande, *"Digital Forensic: The Fascinating World of Digital Evidence"*, Wiley India Pvt. Ltd., 2017.
3. Cory Altheide, Harlan Carvey, *"Digital Forensics with Open Source Tools"*, Syngress Publishing, Inc., 2011.
4. Chris McNab, *"Network Security Assessment"*, O'Reilly Media.
5. Pavan Duggal, *"Cyber Law in India"*, Saakshar Law Publications.
6. Vivek Sood, *"Cyber Law Simplified"*, McGraw Hill Education India.

Suggested Reference Book(s):

1. Clint P. Garrison, *"Digital Forensics for Network, Internet, and Cloud Computing"*, Syngress Publishing, Inc., 2010.
2. Bill Nelson, Amelia Phillips, Christopher Steuart, *"Guide to Computer Forensics and Investigations"*, Cengage Learning, 2014.
3. Debra Littlejohn Shinder, Michael Cross, *"Scene of the Cybercrime: Computer Forensics*

Handbook", 2nd Edition, Syngress Publishing, Inc., 2008.
4. Marjie T. Britz, *"Computer Forensics and Cyber Law"*, Pearson, Third Edition.
5. Justice Yatindra Singh, *"Cyber Laws"*, Universal Law Publishing

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes(2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	**1.72**
CO-2	2	3	3	3	3	1	1	1	2	2	1	**2**
CO-3	2	2	2	2	2	1	1	1	2	2	1	**1.63**
CO-4	2	3	3	3	3	1	1	1	2	3	2	**2.18**
CO-5	2	2	2	2	2	1	1	1	2	2	1	**1.63**
CO-6	2	3	2	3	3	1	1	1	2	3	2	**2.09**
Average	2	2.5	2.5	2.5	2.5	1	1	1	2	2.3	1.5	**1.89**

Social Network Analysis

COURSE CODE: XXXX
COURSE CREDITS: 3
CORE/ELECTIVE: ELECTIVE
L-T-P: 3-0-0

Pre-requisite: Graph Theory, Data Structures and Algorithms, Basics of Machine Learning

Course Objectives:

1. To understand the mathematical foundations of social network analysis including graph-theoretic properties and real-world network characteristics.
2. To apply centrality measures, clustering coefficients, and structural metrics to characterise and compare real-world network datasets.
3. To implement community detection algorithms and model dynamic properties of evolving social networks.
4. To analyse information diffusion, influence propagation, and link prediction mechanisms in large-scale social networks.
5. To design and evaluate Graph Neural Network architectures for node classification, link prediction, and graph-level tasks on social network data.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain the mathematical foundations of graph theory and describe key structural properties — degree distribution, diameter, clustering, and connectivity — of social networks.	Understand
CO-2	Apply centrality measures, network metrics, and graph-theoretic algorithms to characterize, compare, and visualize real-world network datasets.	Apply
CO-3	Implement and evaluate community detection algorithms and analyze temporal dynamics of evolving networks using appropriate models.	Apply
CO-4	Analyze information diffusion, influence propagation models, and link prediction techniques in large-scale directed and weighted networks.	Analyze
CO-5	Design Graph Neural Network architectures — GCN, Graph SAGE, and GAT — for node-level, edge-level, and graph-level prediction tasks on social data.	Create

Course Contents:

Unit	Contents	Lectures Required
1	Introduction to Social Network Analysis: Definition, history, and scope of social network analysis (SNA); Types of networks — directed, undirected,	**4**

	weighted, signed, bipartite, multiplex, and temporal; Real-world network examples — social media (Twitter, Facebook graphs), citation networks, biological networks, collaboration networks, and web graphs; Network data formats and representations — adjacency matrix, adjacency list, edge list, and GraphML; Tools and libraries for SNA — NetworkX, igraph, Gephi, and graph-tool; Introduction to network datasets — SNAP, KONECT, and UCI Network Data Repository.	
2	Graph Theory Foundations for Social Network Analysis: Graph terminology — vertices, edges, paths, walks, trails, and cycles; Connectivity — connected components, strongly and weakly connected components in directed graphs; Degree distribution — Poisson (random graphs) vs. power-law (scale-free networks); Shortest paths — BFS, Dijkstra's algorithm, and the concept of diameter and eccentricity; Clustering coefficient — local (Watts-Strogatz) and global (transitivity); Bipartite graphs — representation, projection, and applications; Graph density, sparsity, and core-periphery structure; Ego networks and structural holes.	**5**
3	Centrality Measures and Network Importance: Degree centrality and normalisation; Closeness centrality and its variants for disconnected graphs; Betweenness centrality — Brandes' algorithm; Eigenvector centrality and the Perron-Frobenius theorem; PageRank — damping factor, iterative computation, and personalised PageRank; HITS (Hyperlink-Induced Topic Search) algorithm — hubs and authorities; Katz centrality and alpha centrality; Structural holes, brokerage, and bridge nodes (Burt's theory); Comparison and visualisation of centrality across real-world networks.	**5**
4	Community Detection in Networks: Definition of community and the modularity (Q) objective function; Hierarchical clustering — agglomerative and divisive approaches; Girvan-Newman algorithm — edge betweenness-based divisive clustering; Louvain algorithm — greedy modularity optimisation; Leiden algorithm — improvements for well-connected communities; Label propagation algorithm (LPA) for scalable community detection; Spectral clustering — graph Laplacian, eigenvector decomposition, and normalised cuts; Overlapping community detection — BigCLAM and DEMON; Community detection in directed, weighted, and bipartite networks; Evaluation metrics — Normalised Mutual Information (NMI), Adjusted Rand Index (ARI), and modularity score.	**6**
5	Network Models and Temporal Network Analysis: Random graph model — Erdős-Rényi G(n, m) and G(n, p); Small-world model — Watts-Strogatz construction and properties; Scale-free model — Barabási-Albert preferential attachment; Forest Fire model and Kroneckergraph model; Stochastic Block Model (SBM) for community-structured graphs; Temporal networks — time-varying graphs, contact sequences, and activity-driven models; Dynamic network analysis — network snapshot aggregation and temporal centrality; Anomaly detection in evolving networks.	**5**
6	Information Diffusion and Influence Maximisation: Epidemic spreading models — SI, SIR, SIS, and SEIR on networks; Threshold-based diffusion — Linear Threshold (LT) model; Probabilistic diffusion — Independent Cascade (IC) model; Influence maximisation problem — NP-hardness	**6**

	proof and greedy approximation; CELF and CELF++ optimisations for scalable influence maximisation; TIM (Two-Phase Influence Maximisation) and IMM algorithms; Rumour spreading, misinformation propagation, and counter-strategy design; Viral marketing, seeding strategies, and real-world campaign analysis; Experimental methods — natural experiments and A/B testing on networks.	
7	Link Prediction and Network Embeddings: Link prediction problem — formulation and evaluation (AUC-ROC, Average Precision); Structural similarity-based methods — Common Neighbours, Jaccard Index, Adamic-Adar, and Resource Allocation; Path-based methods — Katz index and PageRank-based scores; Matrix factorisation for link prediction; Random walk-based graph embeddings — DeepWalk and Node2Vec; LINE (Large-scale Information Network Embedding) for first-order and second-order proximity; Struct2Vec for structural identity-preserving embeddings; Applications of graph embeddings — node classification, link prediction, and graph visualization.	**5**
8	Graph Neural Networks for Social Networks: Limitations of hand-crafted graph features and motivation for deep graph learning; Message-passing framework and the Weisfeiler-Lehman graph isomorphism test; Graph Convolutional Networks (GCN) — spectral and spatial formulations; GraphSAGE — inductive learning with neighbourhood sampling; Graph Attention Networks (GAT) — multi-head attention for node aggregation; Graph Isomorphism Network (GIN) and its expressive power; Scalable GNN training — mini-batch, cluster-GCN, and GraphSAINT sampling; Applications — node classification on citation and social graphs, link prediction, and graph classification; GNN libraries — PyTorch Geometric (PyG) and Deep Graph Library (DGL).	**6**
	Total lectures	**42**

Suggested Text Book(s):

1. David Easley, Jon Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.
2. Stanley Wasserman, Katherine Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.
3. Jure Leskovec, AnandRajaraman, Jeffrey Ullman, Mining of Massive Datasets, Cambridge University Press, 3rd Edition, 2020.
4. William L. Hamilton, Graph Representation Learning, Morgan & Claypool Publishers, 2020.

Suggested Reference Book(s):

1. Mark Newman, Networks: An Introduction, Oxford University Press, 2nd Edition, 2018.
2. Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, Social Media Mining: An Introduction, Cambridge University Press, 2014.
3. Albert-LászlóBarabási, Network Science, Cambridge University Press, 2016.
4. Thomas Kipf, Max Welling, Semi-supervised Classification with Graph Convolutional Networks (Research Paper), ICLR 2017.

Other useful resource(s):

1. NPTEL course — Social Networks: https://nptel.ac.in/courses/106102215
2. NetworkX documentation: https://networkx.org/documentation/stable/

3. PyTorch Geometric tutorials: https://pytorch-geometric.readthedocs.io
4. Stanford SNAP network datasets and tutorials: https://snap.stanford.edu

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	2	2	1	1	1	2	1	2	1.7
CO-2	3	3	2	2	3	1	1	1	2	2	2	2.0
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	3	3	3	3	3	1	1	2	3	2	2	2.4
CO-5	3	3	3	3	3	2	2	2	3	2	3	2.6
Average	**3.0**	**2.8**	**2.6**	**2.4**	**2.8**	**1.2**	**1.2**	**1.4**	**2.4**	**1.8**	**2.2**	**2.2**

Software Construction using Kubernetes and Micro-services

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Operating Systems, Computer Networks, Basic Cloud Computing, Programming (Python/Java).

Course Objectives:

1. To understand the principles of microservices architecture and cloud-native software construction.
2. To learn containerization concepts using Docker and orchestration using Kubernetes.
3. To design scalable, resilient, and distributed systems using microservices.
4. To implement service communication, API gateways, and fault tolerance mechanisms.
5. To deploy and manage applications using Kubernetes in real-world scenarios.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain microservices architecture, cloud-native principles, and containerization concepts.	Understand
CO-2	Apply Docker and Kubernetes concepts for container orchestration and deployment.	Apply
CO-3	Design scalable and fault-tolerant microservices-based systems.	Apply
CO-4	Analyze distributed systems challenges like service communication, load balancing, and failures.	Analyze
CO-5	Develop and deploy real-world applications using Kubernetes clusters.	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Software Construction &Microservices**: Traditional vs Monolithic Architecture, Need for Microservices, Characteristics of Microservices, SOA vs Microservices, 12-Factor App Principles	1
2	**Containerization using Docker**: Introduction to Containers, Virtualization vs Containers, Docker Architecture, Docker Images & Containers, Dockerfile, Docker Compose, Networking in Docker	8
3	**Microservices Design Principles**: Service Decomposition, Database per Service, API Design, RESTful Services, Inter-service Communication (Sync/Async), Service Discovery	3
4	**Kubernetes Fundamentals**: Kubernetes Architecture, Pods, ReplicaSets, Deployments, Services, Namespaces, ConfigMaps& Secrets	3

5	**Advanced Kubernetes**: Auto-scaling (HPA), Rolling Updates & Rollbacks, StatefulSets, Persistent Volumes, Helm Charts	6
6	**Microservices Infrastructure**: API Gateway, Load Balancing, Circuit Breaker, Message Queues (Kafka/RabbitMQ), Service Mesh (Istio basics)	9
7	**CI/CD and Deployment**: DevOps Basics, CI/CD Pipelines, Jenkins/GitHub Actions, Container Registry, Kubernetes Deployment Strategies (Blue-Green, Canary) Authentication & Authorization, OAuth/JWT, Logging & Monitoring (Prometheus, Grafana), Fault Tolerance & Observability	12
Total lectures		**42**

Suggested Text Book(s):

1. Sam Newman, *Building Microservices*, O'Reilly, 2nd Edition
2. Nigel Poulton, *Docker Deep Dive*, 2023 Edition
3. Kelsey Hightower et al., *Kubernetes: Up and Running*, O'Reilly

Suggested Reference Book(s):

1. Chris Richardson, *Microservices Patterns*, Manning Publications
2. Brendan Burns, *Designing Distributed Systems*, O'Reilly
3. Marko Luksa, *Kubernetes in Action*, Manning

Other useful resource(s):

1. NPTEL https://onlinecourses.nptel.ac.in/noc26_cs55/preview
2. Kubernetes Official Documentation: https://kubernetes.io/docs
3. Docker Docs: https://docs.docker.com
4. Microservices.io by Chris Richardson https://microservices.io/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	1	1	2	**1.3**
CO-2	3	3	2	2	3	1	1	2	1	2	2	**1.9**
CO-3	3	2	3	2	3	1	1	2	1	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	1	2	**1.8**
CO-5	3	2	3	2	3	1	1	2	2	2	3	**2.1**
Average	**2.8**	**2.4**	**2.2**	**2**	**2.6**	**1**	**1**	**1.6**	**1.4**	**1.6**	**2.2**	

AI for Finance

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Python Programming, NLP, Blockchain

Course Objectives:

1. Understand AI/ML foundations in finance
2. Analyze financial data using intelligent models
3. Design AI-driven financial systems
4. Evaluate risks, ethics, and future trends

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the structure of financial systems and data driven decision making	Understand
CO-2	Analyze financial data using analytical techniques and visualization	Analyze
CO-3	Apply AI driven approaches in trading, risk, and fraud detection.	Apply
CO-4	Evaluate modern FinTech systems including blockchain and digital finance	Analyze
CO-5	Analyze portfolio strategies using AI based approaches.	Analyze
CO-6	Assess ethical, legal, and security issues in AI driven finance	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Intelligent Finance:** Overview of Financial Systems and Markets, Evolution of Finance: Traditional → Digital → Intelligent, Introduction to Artificial Intelligence in Finance, Role of Data in Financial Decision-Making	4
2	**Foundations of AI & ML:** Basics of Machine Learning (Supervised, Unsupervised, Reinforcement), Data Preprocessing and Feature Engineering, Regression Models in Finance, Classification Techniques (Logistic Regression, SVM), Neural Networks Basics, Deep Learning Overview	6
3	**Financial Data Analytics:** Time Series Analysis in Finance, ARIMA, GARCH Models, Financial Indicators and Technical Analysis, Big Data in Finance (Structured vs Unstructured Data), Data Visualization for Financial Insights	5
4	**Algorithmic Trading & AI:** Introduction to Algorithmic Trading, Trading Strategies using AI, High-Frequency Trading Systems,	5

	Reinforcement Learning in Trading, Backtesting and Performance Metrics	
5	**Risk Management & Fraud Detection:** Financial Risk Types (Market, Credit, Operational), AI in Credit Scoring, Fraud Detection using Machine Learning, Anomaly Detection Techniques, Regulatory Compliance and Risk Modeling	**5**
6	**Natural Language Processing in Finance:** Introduction to NLP in Finance, Sentiment Analysis of Financial News, Text Mining for Market Prediction, Chatbots and Robo-Advisors	**4**
7	**Blockchain & FinTech Innovations:** Introduction to Blockchain Technology, Cryptocurrencies and Smart Contracts, AI in FinTech Ecosystems, Digital Payments and Financial Inclusion	**4**
8	**Portfolio Management using AI:** Portfolio Theory Basics, AI-based Portfolio Optimization, Robo-Advisory Systems, Behavioral Finance and AI	**4**
9	**Ethical, Legal & Security Aspects:** Ethical Issues in AI Finance, Data Privacy and Security, Bias and Fairness in Financial Algorithms	**3**
10	**Future Trends & Case Studies:** Case Studies (Banks, Hedge Funds, FinTech Startups), Future of Intelligent Finance (Quantum Finance, Autonomous Systems)	**2**
	Total lectures	**42**

Suggested Text Book(s):

1. Intelligent Finance : How AI is Transforming the Financial Industryby Kaustuva Pradhan

Suggested Reference Book(s):

2. Artificial Intelligence in Finance – Yves Hilpisch
3. Machine Learning for Asset Managers – Marcos López de Prado
4. Business Intelligence and Analytics – Ramesh Sharda
5. Data Science for Business – Foster Provost
6. Python for Finance – Yves Hilpisch

Other useful resource(s):

1. Coursera: Courses on AI, Machine Learning, and Financial Engineering (e.g., ML by Andrew Ng)
2. edX: Programs in FinTech, Data Science, and Computational Finance
3. Udemy: Practical courses on algorithmic trading, Python for finance

Evaluation Scheme:

S. No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.63**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.09**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.9**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.18**
CO-6	2	2	2	2	2	1	1	1	3	2	2	**1.8**
Average	**2.6**	**2.3**	**2.3**	**2**	**2.5**	**1**	**1**	**1.1**	**2.3**	**1.8**	**2.1**	

Agentic AI

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning, Python Programming, Basic NLP

Course Objectives:

1. To understand intelligent agents and agentic AI systems.
2. To study architectures like symbolic, reactive, and hybrid agents.
3. To analyze planning and reasoning in autonomous agents.
4. To implement LLM-based agents with tools and memory.
5. To design multi-agent systems for real-world problems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain intelligent agents and architectures	Understand
CO-2	Apply planning and reasoning	Apply
CO-3	Implement LLM-based agents	Apply
CO-4	Analyze multi-agent systems	Analyze
CO-5	Design agentic AI systems	Create

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Intelligent Agents:** Definition, types of agents (reactive, deliberative), agent-environment interaction, rationality	**4**
2	**Agent Architectures:** Symbolic, reactive, hybrid architectures; BDI (Belief-Desire-Intention) models	**6**
3	**Planning in Agents:** State-space search, STRIPS, classical planning, hierarchical planning	**7**
4	**Reasoning and Decision Making:** Logical reasoning, probabilistic reasoning, utility-based agents, MDPs (intro)	**6**
5	**LLM-based Agents:** Prompting, tool use, ReAct framework, chain-of-thought, function calling	**7**
6	**Memory and Tool Integration:** Short-term vs long-term memory, vector DBs, API/tool integration	**5**

7	**Multi-Agent Systems:** Coordination, communication protocols, emergent behavior, negotiation	**4**
8	**Applications of Agentic AI:** Autonomous assistants, robotics, research agents, safety & alignment issues	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Artificial Intelligence: A Modern Approach — Stuart Russell & Peter Norvig
2. AI Engineering — Chip Huyen *Health Care and Medical Sciences*

Suggested Reference Book(s):

1. An Introduction to MultiAgent Systems — Michael Wooldridge
2. Speech and Language Processing — Dan Jurafsky& James H. Martin

Other useful resource(s):

1. UC Berkeley https://www.youtube.com/watch?v=CvZDJxd4LKM&list=PLS01nW3RtgoqGkm4UeqNeZLccW-OGc1fJ
2. Coursera: https://www.coursera.org/specializations/agentic-ai-engineering

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	2	2	1	1	1	1	1	1	**1.5**
CO-2	3	3	2	2	3	1	1	1	2	2	1	**2.0**
CO-3	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-4	2	3	2	3	2	1	1	1	2	2	2	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.6**	**2.2**	**2.2**	**2.6**	**1.0**	**1.0**	**1.2**	**2.0**	**1.8**	**1.6**	

Cyber Security and Digital Safety

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic Computer Knowledge

Course Objectives:

1. To understand fundamental concepts of cyber security, threats, and vulnerabilities.
2. To identify and analyze different types of cyber attacks and defense mechanisms.
3. To apply security practices for safe digital communication and data protection.
4. To understand legal, ethical, and social issues in cyber security.
5. To develop awareness of digital safety practices and cyber hygiene.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO-1	Explain basic concepts of cybersecurity, threats, vulnerabilities, and attacks.	Understand
CO-2	Apply various security mechanisms to protect systems and networks.	Apply
CO-3	Analyze different types of cyber attacks and mitigation techniques.	Analyze
CO-4	Evaluate ethical, legal, and social issues related to cyber security.	Evaluate
CO-5	Design strategies for ensuring digital safety and cyber hygiene practices.	Create

Course Contents:

Unit	**Contents**	**Lectures required**
1	Introduction to Cyber Security: Definition, Importance of Cyber Security, Cyber Threats, Vulnerabilities and Risks, CIA Triad (Confidentiality, Integrity, Availability), Overview of Cyber Attacks, Cyber Security Models and Frameworks	**6**
2	Types of Cyber Threats and Attacks: Malware (Virus, Worm, Trojan, Ransomware), Spyware, Adware, Phishing, Spear Phishing, Social Engineering, Password Attacks, DoS and DDoS Attacks, Man-in-the-Middle Attack, Insider Threats	**6**
3	Network Security: Basics of Network Security, Firewalls, IDS/IPS, VPN, Secure Communication Protocols (HTTPS, SSL/TLS), Wireless Security Issues, Wi-Fi Security Standards, Network Monitoring Tools	**6**
4	Cryptography and Data Security: Basics of Cryptography, Symmetric and Asymmetric Encryption, Hash Functions, Digital Signatures, Public Key Infrastructure (PKI), Key Management, Applications of	**6**

	Cryptography	
5	System Security: Operating System Security, Access Control, Authentication and Authorization, Multi-factor Authentication, Secure Software Development Practices, Patch Management, Endpoint Security	**6**
6	Cyber Laws and Ethics: IT Act, Cyber Laws in India, Cyber Crimes and Legal Remedies, Ethical Hacking, Privacy and Data Protection, Intellectual Property Rights, Digital Rights and Responsibilities	**6**
7	Digital Safety and Cyber Hygiene: Safe Internet Practices, Secure Password Management, Social Media Safety, Email Security, Data Backup and Recovery, Awareness of Cyber Frauds, Cyber Safety Case Studies, Emerging Trends in Cyber Security	**6**
Total lectures		**42**

Suggested Text Book(s):

1. Charles P. Pfleeger, Security in Computing, Pearson Education
2. William Stallings, Cryptography and Network Security, Pearson
3. Nina Godbole &SunitBelapure, Cyber Security: Understanding Cyber Crimes, Wiley

Suggested Reference Book(s):

1. Michael Whitman & Herbert Mattord, Principles of Information Security, Cengage
2. Joseph Migga Kizza, Guide to Computer Network Security, Springer
3. Matt Bishop, Computer Security: Art and Science, Pearson

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.1**
CO-3	3	2	2	3	3	1	1	1	2	2	2	**2.0**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.2**
Average	**2.8**	**2.4**	**2.4**	**2.2**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.2**	

Business Intelligence and Analytics

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge ofmathematics and statistics

Course Objectives:

1. To understand the fundamentals of Business Intelligence (BI) and its role in data-driven decision-making.
2. To learn the data lifecycle (data to information to knowledge) and different types of analytics: descriptive, predictive, and prescriptive.
3. To gain knowledge of data types, sources, and preprocessing techniques, including data cleaning and transformation.
4. To apply basic statistical and analytical methods such as correlation, regression, classification, and clustering.
5. To develop skills in computational tools for analytics, including Python basics, data handling, and visualization.
6. To understand BI systems and technologies, including ETL, Big Data, cloud analytics, and applications of AI/ML.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain fundamental concepts of Business Intelligence and Analytics.	Understand
CO-2	Identify data types and understand data preparation processes.	Understand
CO-3	Apply analytical techniques such as regression, classification, clustering.	Apply
CO-4	Interpret analytical results and derive meaningful insights for business decision making using case-based reasoning.	Apply
CO-5	Analyze real-world problems using data-driven approaches.	Analyze
CO-6	Evaluate real-world business scenarios and propose data driven solutions using analytical thinking	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to BI and Data Driven Systems:** Introduction to Business Intelligence, importance in modern organizations, Evolution from data to information to knowledge, Data driven decision making and decision support systems, Types of analytics: descriptive, predictive, prescriptive, Role of computing systems in analytics and	8

	decision making, Applications of BI across industries (finance, healthcare, retail)	
2	**Data Fundamentals and Processing:** Types of data: structured, semi-structured, unstructured, Sources of data: enterprise systems, web, sensors, Data collection methods and challenges, Data quality issues: missing data, inconsistency, noise, Data cleaning and preprocessing concepts, Data transformation and normalization (conceptual), Introduction to spreadsheets for data handling, Case discussion on data preparation in real systems	**9**
3	**Analytical Techniques:** Introduction to statistics for analytics, Measures of central tendency and variability, Correlation and interpretation, Regression concepts and applications, Classification: concept and real-world use, Classification techniques overview (decision trees, basic models), Clustering: concept and grouping of data, Clustering techniques overview (k-means concept), Trend analysis and forecasting basics, Case studies in analytics (business decision scenarios)	**10**
4	**Computational Foundations for Analytics:** Introduction to programming in analytics (Python overview), Data handling concepts using Python (Pandas idea), Data visualization concepts and dashboards, Overview of tools (Power BI, Tableau) and their role	**8**
5	**BI Systems and Technologies:** ETL process: Extract, Transform, Load, Introduction to Big Data concepts (volume, velocity, variety) Cloud based analytics systems (conceptual overview), AI and machine learning in analytics (applications)	**7**
Total lectures		**42**

Suggested Text Book(s):

3. Efraim Turban, Ramesh Sharda, DursunDelen – Business Intelligence: A Managerial Perspective on Analytics.
4. Jiawei Han, Micheline Kamber, Jian Pei – *Data Mining: Concepts and Techniques*
5. Foster Provost, Tom Fawcett – *Data Science for Business*

Suggested Reference Book(s):

1. G. K. Gupta – *Introduction to Data Mining with Case Studies*
2. S. Sumathi, S. N. Sivanandam – *Data Mining and Data Warehousing*

Other useful resource(s):

1. Business Intelligence (BI) Essentials Course https://www.coursera.org/learn/business-intelligence-essentials?

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	2	1	1	1	1	1	1	1	1	1	1	**1.1**
CO-2	2	2	1	1	1	1	1	1	1	1	1	**1.2**
CO-3	3	2	1	2	2	1	1	1	1	1	1	**1.5**
CO-4	2	3	1	2	1	1	1	1	2	1	1	**1.5**
CO-5	2	3	2	2	1	1	1	1	1	1	1	**1.5**
CO-6	2	3	3	2	2	2	1	1	2	1	2	**1.9**
Average	**2.2**	**2.3**	**1.5**	**1.7**	**1.3**	**1.2**	**1.0**	**1.0**	**1.3**	**1.0**	**1.2**	

User Interface and User Experience Design

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basic knowledge of computers and web technologies

Course Objectives:

1. To understand fundamental UI/UX design principles and methodologies.
2. To develop skills in modern design tools and AI-assisted workflows.
3. To apply user-centered design, usability, and accessibility concepts.
4. To design wireframes, layouts, and prototypes for real-world applications.
5. To evaluate user experience through testing, feedback, and iteration.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain UI/UX concepts, principles, and usability fundamentals.	Understand
CO-2	Apply design tools and techniques to create UI components and layouts.	Apply
CO-3	Analyze user behavior and design effective navigation and structures.	Analyze
CO-4	Evaluate designs using accessibility standards and usability testing.	Evaluate
CO-5	Design complete UI/UX solutions including wireframes and prototypes.	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Design & UI/UX Fundamentals: What is design, purpose & problem solving, research & ideation, prototyping, UI vs UX, user-centered design, usability, trends	**6**
2	Design Tools & AI Integration: Overview of tools like Figma, Adobe XD, Sketch, AI tools, prompt crafting, plugins, workflow efficiency	**6**
3	Structure, Navigation & UX Foundations: Information architecture, user flows, content organization, usability enhancements, 5 Elements of UX (Strategy, Scope, Structure, Skeleton, Surface)	**6**
4	Layout, Typography & Color Theory: Grid systems, alignment, spacing, visual hierarchy, responsive design, typography, readability, contrast, color theory, accessibility	**6**
5	UI Components & Visual Elements: Icons, images, storytelling, forms, inputs, validation, usability, accessibility in UI components	**6**

6	Accessibility, Design Principles & UX Writing: WCAG principles, usability, balance, contrast, simplicity, consistency, microcopy, tone & voice, inclusive design, cognitive load reduction	**6**
7	UX Process, MVP & Real-world Application: UX design process (research, ideation, prototyping, testing), MVP concept, user validation, iteration, optimization, case studies	**6**
Total lectures		**42**

Suggested Text Book(s):

1. Don't Make Me Think – Steve Krug
2. The Design of Everyday Things – Don Norman
3. Lean UX – Jeff Gothelf

Suggested Reference Book(s):

1. Hooked – Nir Eyal
2. 100 Things Every Designer Needs to Know About People – Susan Weinschenk

Other useful resource(s):

1. Nielsen Norman Group – UX articles and case studies: https://www.nngroup.com/
2. Figma – UI/UX design practice platform: https://www.figma.com/
3. Interaction Design Foundation – UI/UX learning

   resources: https://www.interaction-design.org/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	2	1	2	1	1	1	2	1	2	1.6
CO-2	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-3	3	2	3	2	3	1	1	1	2	2	2	2.0
CO-4	2	3	2	3	2	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	2	3	2	2	2.2
Average	**2.8**	**2.4**	**2.6**	**2.0**	**2.6**	**1.0**	**1.0**	**1.2**	**2.2**	**1.8**	**2.0**	

Software Testing Methodologies

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Software Engineering

Course Objectives:

1. To understand the basic testing procedures.
2. To able to support in generating test cases and test suites.
3. To able to test the applications manually by applying different testing methods and automation tools.
4. To apply tools to resolve the problems in Real time environment.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain fundamental concepts of software testing including purpose, models, bug taxonomy, and consequences of bugs.	Understand
CO-2	Apply structural testing techniques such as path testing, flow graphs, transaction flow testing, and dataflow testing to identify defects in software systems.	Apply
CO-3	Analyze domain testing, syntax testing, and logic-based testing techniques for designing effective test cases.	Analyze
CO-4	Evaluate state-based and graph-based testing approaches including state graphs, transition testing, graph matrices, and their applications.	Evaluate
CO-5	Utilize software testing tools and automation techniques (e.g., WinRunner, LoadRunner, JMeter) for test case execution, debugging, and result analysis.	Apply

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Purpose of Testing, Dichotomies, Model for Testing, Consequences of Bugs, Taxonomy of Bugs. **Flow graphs and Path testing:** Basics Concepts of Path Testing, Predicates, Path Predicates and Achievable Paths, Path Sensitizing, Path Instrumentation, Application of Path Testing.	**6**
2	**Transaction Flow Testing:** Transaction Flows, Transaction Flow Testing Techniques. **Dataflow testing:** Basics of Dataflow Testing, Strategies in Dataflow Testing, Application of Dataflow Testing.	**6**
3	**Domain Testing:** Domains and Paths, Nice & Ugly Domains, Domain testing, Domains and Interfaces Testing, Domain and Interface Testing, Domains and Testability.	**7**

	Paths, Path products and Regular expressions: Path Products & Path Expression, Reduction Procedure, Applications, Regular Expressions & Flow Anomaly Detection.	
4	**Syntax Testing:** Why, What and How, A Grammar for formats, Test Case Generation, Implementation and Application and Testability Tips. **Logic Based Testing:** Overview, Decision Tables, Path Expressions, KV Charts, and Specifications.	**7**
5	**State, State Graphs and Transition Testing:** State Graphs, Good & Bad State Graphs, State Testing, and Testability Tips. **Graph Matrices and Application:** Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm.	**7**
6	**Software Testing Tools:** Introduction to Testing, Automated Testing, Concepts of Test Automation, Introduction to list of tools like Win runner, Load Runner, Jmeter, About Win Runner, Using Win runner, Mapping the GUI, Recording Test, Working with Test, Enhancing Test, Checkpoints, Test Script Language, Putting it all together, Running and Debugging Tests, Analyzing Results, Batch Tests, Rapid Test Script Wizard.	**9**
	Total lectures	**42**

Suggested Text Book(s):

1. Software testing techniques – Boris Beizer, Dreamtech, second edition.
2. Software Testing- Yogesh Singh, Camebridge

Suggested Reference Book(s):

1. The Craft of software testing - Brian Marick, Pearson Education.
2. Software Testing, 3rd edition, P.C. Jorgensen, Aurbach Publications (Dist.by SPD).
3. Software Testing, N.Chauhan, Oxford University Press.
4. Introduction to Software Testing, P.Ammann&J.Offutt, Cambridge Univ.Press.
5. Effective methods of Software Testing, Perry, John Wiley, 2nd Edition, 1999.
6. Software Testing Concepts and Tools, P.NageswaraRao, dreamtech Press
7. Win Runner in simple steps by Hakeem Shittu, 2007 Genixpress.
8. Foundations of Software Testing, D.Graham& Others, Cengage Learning.

Other useful resource(s):

3. **Link to NPTEL course contents:**https://nptel.ac.in/courses/106101061
4. **Link to NPTEL course contents:** https://nptel.ac.in/courses/106104128

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	1	2	1	1	1	1	1	2	**1.5**
CO-2	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-3	3	3	2	3	2	1	1	1	1	1	2	**1.8**
CO-4	3	3	3	3	2	1	1	1	2	1	3	**2.1**
CO-5	3	2	2	2	3	1	1	2	2	2	3	**2.1**
Average	**3**	**2.6**	**2**	**2.2**	**2.4**	**1**	**1**	**1.2**	**1.4**	**1.2**	**2.4**	

Social Media Analytics

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basics of Machine Learning, Python Programming, Basic NLP

Course Objectives:

1. To understand the key concepts in social media data and data analytics.
2. To study how to collect data from social media platforms.
3. To study, analyze and visualize real-world social media data from various sources.
4. To understand the intersection of social media platforms and network science.
5. To design case studies from real life datasets.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Explain data analytics for social media analytics	Understand
CO-2	Apply data science fundamentals for collecting and processing social media data	Apply
CO-3	Implement data exploration and visualization on social media data	Apply
CO-4	Analyze social media data as network graphs	Analyze
CO-5	Design sentiment analysis, user behavior analysis and cross-cultural studies	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Data Analytics: Data in the Real World, Structured vs. Unstructured Data, Analyzing Structured Data, Visualization of Data	**4**
2	Collecting and Extracting Social Media Data: Introduction to Python Programming, Using Python to Extract Data from Twitter/X, Using Python to Extract Data from YouTube	**8**
3	Introduction to Data Analysis, Visualization, and Exploration: Analyzing Social Media Data Using Python, Introduction to R, Social Media Data Analysis with R	**7**
4	An Introduction to Network Analysis with Gephi	**5**

5	Case Study 1: Sentiment Analysis with Twitter/X Data, Text Mining of Twitter/X Data, Unstructured Text Analytics in Practice: Sentiment & Concept Exploration	**5**
6	Case Study 2: User Behavior Analysis with Twitter/X Data, User Profile Mining of Twitter/X Data,Behavioral pattern mining	**7**
7	Case Study 3: Cross-cultural studies on social media networks, significance and challenges.	**6**
Total lectures		**42**

Suggested Text Book(s):

1. **Data Analytics for Social Microblogging Platforms, First Edition**
DebabrataSamanta, Soumi Dutta, Asit Kumar Das, Saptarshi Ghosh
(Publisher: Elsevier Science and Technology Books)
https://shop.elsevier.com/books/data-analytics-for-social-microblogging-platforms/dutta/978-0-323-91785-8
2. **Social Media and Web Analytics: Turning Insights into Action in A Digital World**
Mukul Gupta, Deepa Gupta, Parth Mukul Gupta
(Publisher: PHI Learning) (ISBN: 9789354439605)
https://www.phindia.com/Books/BookDetail/9789354439605/SOCIAL-MEDIA-AND-WEB-ANALYTICS-GUPTA-GUPTA-GUPTA

Suggested Reference Book(s):

1. Mining Social Media: Finding Stories in Internet Data
Lam Thuy Vo(Publisher: No Starch Press)(ISBN-13: 9781593279165)
https://nostarch.com/miningsocialmedia
2. Sentiment Analysis in Social Networks
Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, Bing Liu
(Publisher: Elsevier Science and Technology Books) (ISBN: 978-0-12-804412-4)
https://www.sciencedirect.com/book/edited-volume/9780128044124/sentiment-analysis-in-social-networks

Other useful resource(s):

1. Coursera: Social Media Data Analytics
https://www.coursera.org/learn/social-media-data-analytics
2. Network Science, 2021
https://www.youtube.com/watch?v=1T5-BG6yngM&list=PLriUvS7IljvkGesFRuYjqRz4lKgodJgh2

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	2	1	2	2	1	1	1	1	1	1	1.5
CO-2	3	3	2	2	3	1	1	1	2	2	1	2.0
CO-3	3	3	3	2	3	1	1	1	2	2	2	2.1
CO-4	2	3	2	3	2	1	1	1	2	2	2	2.0
CO-5	3	2	3	2	3	1	1	2	3	2	2	2.2
Average	2.8	2.6	2.2	2.2	2.6	1.0	1.0	1.2	2.0	1.8	1.6	

Digital Twin - Fundamental Concepts to Applications in Advanced Manufacturing

COURSE CODE: XXXXXXXX

COURSE CREDITS: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Machine Learning, Artificial Intelligence

Course Objectives:

1. Learn about fundamental ideas about sensor electronics and data acquisition
2. Provide good understanding of how signal and image processing techniques can be used in digital replica of physical process.
3. To understand high-level overview of digital twins and theirunder utilization in the field of asset management and maintenance.
4. To implement artificial intelligence and machine learning for decision making.
5. To explore the necessity for the practical application of Digital Twin in Industry.
6. Deploys developed digital twins on Amazon Web Services.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Describe the role of Digital Twin in the modern era of manufacturing.	Understand
CO-2	Sensor dynamics and signal conversion in sensor electronics.	Understand
CO-3	Signal processing and image processing for Digital Twin.	Apply
CO-4	Data communication: Edge, Fog, and Cloud computing.	Understand
CO-5	Artificial Intelligence and Machine Learning in manufacturing.	Apply
CO-6	Digital Twin applications in terms of prototypes.	Apply

Course Contents:

Unit	Contents	Lectures required
1	Evolution of Manufacturing and Its Journey Towards Digital Twin: A Few Fascinating Facts, Manufacturing and the Industrial Revolution, Instruction set design, Introduction to Digital Twin, Key Aspects of a Digital Twin Model, Components for Envisaging the Digital Twin, Utilities of the Digital Twin in Manufacturing	**4**
2	Sensor Electronics for Digital Twin: The Need for Electronics in Manufacturing, Introduction to Sensor and Transducer, Role of a Sensor in Manufacturing, Types of Sensors, A Guide on Sensors for Digital Twins, Performance Indices of a Sensor	**4**

3	Signal Processing for Digital Twin: Introduction to Signal, Meaning of the Term 'Signal', Signal as Indirect Means of Monitoring, Importance of Signal Processing in Digital Twin, Signal Acquisition and Its Features, Signal Conditioning, Analogue-to-Digital Converter, The Sampling Rate of a Signal, Signal Acquisition, Arduino Microcontroller Input/Output Module and PLC for Industrial Applications, Noise in Signal, Methods of Signal Processing	**6**
4	Image Processing for Digital Twin: Selection of Process Zone or Application Zone, Image Acquisition, Image Enhancement, Image Segmentation, Feature Extraction and Object Recognition, Texture Analyses, Summary of Various Image Processing Techniques Used for Real-Time Process Control and Inspection in Manufacturing	**6**
5	Data Communication – Edge, Fog, and Cloud Computing: IoT and Network, IoT Framework, Introduction to the Edge, Fog, and Cloud Computing, The Necessity of Cloud and Edge Computing in Industry 4.0 Perspective, Edge Versus Cloud Computing, Application Classification, Data Communication Technologies, Network Architectures for Edge/Cloud Computing, Real-Life Example in Manufacturing, Futuristic Concept – 5G in Manufacturing	**10**
6	Artificial Intelligence and Machine Learning in Manufacturing: Introduction to Artificial Intelligence, Foundation of Artificial Intelligence, Requirement of Artificial Intelligence in a Digital Twin, Deep into Artificial Intelligence – The Knowledge Pyramid, Sensor Signal Processing, Machine Learning, Artificial Neural Network, Adaptive Filtering, Deep Learning, Transfer Learning, Automation of Sensor Data Fusion and Sensor Selection, Automation of Compression and Partitioning	**6**
7	Digital Twin Application: Concept of Digital Twin, Realizing a Digital Twin Model, Digital Twin as a Tool Throughout the Life Cycle, Digital Twins for Design and Manufacturing, Digital Twin for Iron and Steel Product Life Cycle, Digital Twin for Weld-Joint Life Cycle, Digital Twins for Service, Challenges and Intelligence, Real-Life Examples in Manufacturing	**6**
	Total lectures	**42**

Suggested Text Book(s):

1. Pal, SurjyaKanta, Debasish Mishra, Arpan Pal, Samik Dutta, Debashish Chakravarty, and Srikanta Pal. "Digital Twin – Fundamental Concepts to Applications in Advanced Manufacturing." (2021).
2. Khaled, Nassim, BibinPattel, and Affan Siddiqui. *Digital Twin Development and Deployment on the Cloud: Developing Cloud-Friendly Dynamic Models Using Simulink®/SimscapeTM and Amazon AWS.* Academic Press, 2020.

Other useful resource(s):

1. https://www.elsevier.com/books/practical-design-and-application-of-model-predictive-control/khaled/978-0-12-813918-9
2. https://www.springer.com/gp/book/9781447123293

3. https://www.mathworks.com/discovery/digital-twin.html
4. https://aws.amazon.com/marketplace/pp/Bosch-Software-Innovations-Bosch-IoT-Things/B07DTJK8MV
5. https://www.comsol.com/
6. https://www.simscale.com/
7. https://www.anylogic.com/
8. https://www.mathworks.com/

Evaluation Scheme:

S. No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2)-10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**Average**
CO-1	3	2	2	1	2	1	1	1	2	1	2	**1.6**
CO-2	3	3	3	2	3	1	1	1	2	2	2	**2.0**
CO-3	3	2	2	2	3	1	1	1	2	2	2	**1.8**
CO-4	2	3	2	3	2	1	1	1	2	2	3	**2.0**
CO-5	3	2	3	2	3	1	1	2	3	2	2	**2.1**
Average	2	2	2	2	2	1	1	1	3	2	2	