BTECH COMPUTER SCIENCE AND ENGINEERING SPECIALIZATION IN CYBER SECURITY

COURSE STRUCTURE

EFFECTIVE: 2026-27 ADMISSION BATCH

Approved in Academic Council held on 30 July 2025

B.TECH. COMPUTER SCIENCE AND ENGINEERING

PROGRAM OBJECTIVES

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

Approved in Academic Council held on 30 July 2025

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

FIRST SEMESTER

S. NO.	COURSE			CONTACT HOURS				CREDITS
	CATEGORY	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	
1	BSC	25B11MA113	MATHEMATICS-I	3	1	0	4	4
2	BSC	25B11PH111	PHYSICS-I	3	1	0	4	4
3	BSC	25B17PH171	PHYSICS LAB-I	0	0	2	2	1
4	ESC	25B11CI11	SOFTWARE DEVELOPMENT FUNDAMENTALS-I	3	1	0	4	4
5	ESC	25B17CI172	SOFTWARE DEVELOPMENT FUNDAMENTALS LAB-I	0	0	2	2	1
6	ESC	25B11EC111	BASIC ELECTRONICS	3	1	0	4	4
7	ESC	25B17EC171	BASIC ELECTRONICS LAB	0	0	2	2	1
8	HSC	25B11HS111	ENGLISH	1	0	2	3	2
9	ESC	25B17GE171	WORKSHOP	0	0	3	3	1.5
			TOTAL				**28**	**22.5**

SECOND SEMESTER

S. NO.	COURSE			CONTACT HOURS				CREDITS
	CATEGORY	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	
1	BSC	25B11MA21	MATHEMATICS-II	3	1	0	4	4
2	BSC	25B11PH211	PHYSICS-II	3	1	0	4	4
3	BSC	25B17PH271	PHYSICS LAB-II	0	0	2	2	1
4	ESC	25B11CI211	SOFTWARE DEVELOPMENT FUNDAMENTALS-II	3	1	0	4	4
5	ESC	25B17CI271	SOFTWARE DEVELOPMENT FUNDAMENTALS LAB-II	0	0	2	2	1
6	HSC	25B17HS271	LIFE SKILLS & PROFESSIONAL COMMUNICATION LAB	0	0	2	2	QUALIFYING
7	ESC	25B17GE172	ENGINEERING DRAWING & DESIGN	0	0	3	3	1.5
8	HSC	25B11HS211	UNIVERSAL HUMAN VALUES (UHV)	2	1	0	3	3
			TOTAL				**24**	**18.5**

THIRD SEMESTER

S. NO.	COURSE			CONTACT HOURS				CREDITS
	CATEGORY	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	
1	BSC		MATHEMATICAL FOUNDATIONS FOR ARTIFICIAL INTELLIGENCE AND DATA SCIENCE	3	1	0	4	4
2	PCC		THEORY OF COMPUTATION	3	0	0	3	3
3	PCC		DATA STRUCTURES	3	1	0	4	4
4	PCC		DATA STRUCTURES LAB	0	0	2	2	1
5	PCC		DATABASE MANAGEMENT SYSTEMS	3	0	0	3	3
6	PCC		DATABASE MANAGEMENT SYSTEMS LAB	0	0	2	2	1
7	PCC		UNIX PROGRAMMING LAB	1	0	2	3	2
8	PCC		OBJECT ORIENTED PROGRAMMING USING JAVA	0	0	2	2	1
9	HSC		ECONOMICS	2	1	0	3	3
10	PRC		SUMMER TRAINING-I (4 WEEKS)*	0	0	0	0	2
11	PCC		COMPETITIVE PROGRAMMING-I	0	0	2	2	1
TOTAL							**28**	**25**
SPECIALIZATION IN CYBER SECURITY (CS)								
12	PCC		INFORMATION AND CYBER SECURITY FOUNDATIONS	3	0	0	3	3
13	PCC		INFORMATION AND CYBER SECURITY FOUNDATIONS LAB	0	0	2	2	1
TOTAL							**33**	**29**

Approved in Academic Council held on 30 July 2025

FOURTH SEMESTER

S. NO.	COURSE			CONTACT HOURS				CREDITS
	CATEGORY	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	
1	HSC		HSS ELECTIVE – 1	2	1	0	3	3
2	PCC		DIGITAL SYSTEMS AND COMPUTER ORGANISATION	3	0	0	3	3
	PCC		DIGITAL SYSTEMS AND COMPUTER ORGANISATION LAB	0	0	2	2	1
3	PCC		DESIGN AND ANALYSIS OF ALGORITHMS	3	1	0	4	4
4	PCC		DESIGN AND ANALYSIS OF ALGORITHMS LAB	0	0	2	2	1
5	PCC		ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING	3	0	0	3	3
6	PCC		ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LAB	0	0	2	2	1
7	PCC		SOFTWARE ENGINEERING	3	0	0	3	3
8	PCC		COMPETITIVE PROGRAMMING-II	0	0	2	2	1
9	PEC		DISCIPLINE ELECTIVE – 1	2	0	0	2	2
10	PEC		DISCIPLINE ELECTIVE – 1 LAB	0	0	2	2	1
11	OMC		ENVIRONMENTAL STUDIES	3	0	0	3	QUALI FYING
TOTAL							**31**	**23**
SPECIALIZATION IN CYBER SECURITY (CS)								
12	PCC		ARTIFICIAL INTELLIGENCE: RECENT TRENDS AND APPLICATIONS	3	0	0	3	3
13	PCC		ARTIFICIAL INTELLIGENCE: RECENT TRENDS AND APPLICATIONS LAB	0	0	2	2	1
TOTAL							**36**	**27**

Approved in Academic Council held on 30 July 2025

FIFTH SEMESTER

S. NO.	COURSE			CONTACT HOURS				CREDITS
	CATEGORY	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	
1	PCC		OPERATING SYSTEMS	3	0	0	3	3
2	PCC		OPERATING SYSTEMS LAB	0	0	2	2	1
3	PCC		COMPUTER NETWORKS	3	1	0	4	4
4	PCC		COMPUTER NETWORKS LAB	0	0	2	2	1
5	PCC		FULL STACK DEVELOPMENT LAB	0	0	2	2	1
6	PEC		DISCIPLINE ELECTIVE – 2	2	0	0	2	2
7	PEC		DISCIPLINE ELECTIVE – 2 LAB	0	0	2	2	1
8	PEC		DISCIPLINE ELECTIVE – 3	2	0	0	2	2
9	PEC		DISCIPLINE ELECTIVE – 3 LAB	0	0	2	2	1
10	BSC		SCIENCE ELECTIVE	3	0	0	3	3
11	OMC		INDIAN CONSTITUTION & TRADITIONAL KNOWLEDGE	3	0	0	3	QUALIFYING
12	PRC		SUMMER TRAINING-II (6 WEEKS) *	0	0	0	0	2
13	PCC		COMPETITIVE PROGRAMMING-III	0	0	2	2	1
14	HSC		LOGICAL AND QUANTITATIVE TECHNIQUES-I	2	0	0	2	2
TOTAL							**31**	**24**
SPECIALIZATION IN CYBER SECURITY (CS)								
15	PCC		CLOUD COMPUTING AND SECURITY	3	0	0	3	3
16	PCC		CLOUD COMPUTING AND SECURITY LAB	0	0	2	2	1
TOTAL							**36**	**28**

SIXTH SEMESTER

S. NO.	COURSE			CONTACT HOURS				CREDITS
	CATEGORY	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	
1	PCC		WEB TECHNOLOGY	3	0	0	3	3
2	PCC		WEB TECHNOLOGY LAB	0	0	2	2	1
3	PCC		ADVANCED DATA STRUCTURES AND ALGORITHMS	3	0	0	3	3
4	PCC		ADVANCED DATA STRUCTURES AND ALGORITHMS LAB	0	0	2	2	1
5	PCC		DISTRIBUTED AND CLOUD COMPUTING **OR** INFORMATION SECURITY AND CRYPTOGRAPHY	3	0	0	3	3
6	PEC		DISCIPLINE ELECTIVE – 4	3	0	0	3	3
7	PEC		DISCIPLINE ELECTIVE – 5	3	0	0	3	3
8	OEC		OPEN ELECTIVE – 1	2	0	0	2	2
9	VALUE ADDED		SELECTED VALUE-ADDED COURSE	2	0	0	2	AUDIT
10	HSC		SOFT SKILLS FOR EMPLOYABILITY	0	0	2	2	1
11	PRC		MINOR PROJECT	0	0	4	4	2
12	HSC		LOGICAL AND QUANTITATIVE TECHNIQUES-II	2	0	0	2	2
TOTAL							**31**	**24**
SPECIALIZATION IN CYBER SECURITY (CS)								
13	PCC		NETWORK SECURITY AND CRYPTOGRAPHY	3	0	0	3	3
14	PCC		DIGITAL FORENSICS	3	0	0	3	3
TOTAL							**37**	**30**

Approved in Academic Council held on 30 July 2025

SEVENTH SEMESTER

S. NO.	COURSE			CONTACT HOURS				CREDITS
	CATEGORY	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	
1	PEC		DISCIPLINE ELECTIVE – 6	3	0	0	3	3
2	OEC		OPEN ELECTIVE – 2	3	0	0	3	3
3	PRC		MAJOR PROJECT PART – 1	0	0	0	8	4
4	PRC		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		DISCIPLINE ELECTIVE –7	3	0	0	3	3
2	OEC		OPEN ELECTIVE –3	3	0	0	3	3
3	PRC		MAJOR PROJECT PART –2	0	0	0	16	8
TOTAL							**22**	**14**

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

Total Program Credits:22.5 + 18.5 + 29 + 27 + 28 + 30 + 14 + 14 = 183

Bucket Wise Tentative List of Discipline Electives **(To be updated time to time)**

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

S. NO.	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	CREDITS
1		DATA ANALYTICS USING R AND PYTHON	2	0	0	2	2
2		MOBILE APPLICATION DEVELOPMENT	2	0	0	2	2
3		FUNDAMENTALS OF SMART SYSTEMS AND IOT	2	0	0	2	2
4		INTRODUCTION TO COMPILER DESIGN	2	0	0	2	2
5		DATA ANALYTICS USING R AND PYTHON LAB	0	0	2	2	1
6		MOBILE APPLICATION DEVELOPMENT LAB	0	0	2	2	1
7		FUNDAMENTALS OF SMART SYSTEMS AND IOT LAB	0	0	2	2	1
8		INTRODUCTION TO COMPILER DESIGN LAB	0	0	2	2	1

Approved in Academic Council held on 30 July 2025

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

S. NO.	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	CREDITS
1		FUNDAMENTALS OF SOFT COMPUTING	2	0	0	2	2
2		FUNDAMENTALS OF COMPUTER AND CYBER SECURITY	2	0	0	2	2
3		DATA MINING AND DATA WAREHOUSING	2	0	0	2	2
4		AGILE SOFTWARE DEVELOPMENT PROCESS	2	0	0	2	2
5		IOT ANALYTICS	2	0	0	2	2
6		FUNDAMENTALS OF SOFT COMPUTING LAB	0	0	2	2	1
7		FUNDAMENTALS OF COMPUTER AND CYBER SECURITY LAB	0	0	2	2	1
8		DATA MINING AND DATA WAREHOUSING LAB	0	0	2	2	1
9		AGILE SOFTWARE DEVELOPMENT PROCESS LAB	0	0	2	2	1
10		IOT ANALYTICS 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		IMAGE PROCESSING AND COMPUTER VISION	2	0	0	2	2
2		INTRODUCTION TO BLOCKCHAIN TECHNOLOGY	2	0	0	2	2
3		COMPUTING FOR DATA SCIENCE	2	0	0	2	2
4		SENSOR TECHNOLOGY & ANDROID PROGRAMMING	2	0	0	2	2
5		CONCEPT OF GRAPH THEORY	2	0	0	2	2
6		IMAGE PROCESSING AND COMPUTER VISION LAB	0	0	2	2	1
7		INTRODUCTION TO BLOCKCHAIN TECHNOLOGY LAB	0	0	2	2	1
8		COMPUTING FOR DATA SCIENCE LAB	0	0	2	2	1
9		SENSOR TECHNOLOGY & ANDROID PROGRAMMING LAB	0	0	2	2	1
10		CONCEPT OF GRAPH THEORY 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	18B12CS428	INTRODUCTION TO DEEP LEARNING	3	0	0	3	3
2	22B12CS419	CRYPTOCURRENCY TECHNOLOGIES	3	0	0	3	3
3	16B1NCI648	INFORMATION RETRIEVAL AND SEMANTIC WEB	3	0	0	3	3
4	22B12CS422	CLOUD COMPUTING ESSENTIALS: AZURE AND AWS	3	0	0	3	3

Discipline Elective – 5 (offered in 6th Semester)

S. NO.	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	CREDITS
1	21B12CS417	MACHINE LEARNING AND BIG DATA	3	0	0	3	3
2	21B12CS415	SECURE DESIGN OF SOFTWARE SYSTEMS	3	0	0	3	3

3	21B12CS413	FOG AND EDGE COMPUTING	3	0	0	3	3

Discipline Elective – 6 (offered in 7th Semester)

S. NO.	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	CREDITS
1	17B1NCI731	MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING	3	0	0	3	3
2	21B12CS418	ETHICAL HACKING & PREVENTION	3	0	0	3	3
3	21B12CS314	INTRODUCTION TO LARGE SCALE DATABASE SYSTEMS	3	0	0	3	3
4	19B12CS427	INTRODUCTION TO DEVOPS	3	0	0	3	3
5	22B12CS411	INDUSTRIAL AUTOMATION AND IOT	3	0	0	3	3

Discipline Elective – 7 (offered in 8th Semester)

S. NO.	COURSE CODE	COURSE TITLE	L	T	P	TOTAL	CREDITS
1	22B12CS415	AI FOR HEALTHCARE & SMART SYSTEMS	3	0	0	2	3
2	22B12CS412	DIGITAL FORENSICS AND CYBER LAWS	3	0	0	2	3
3	15B1NCI732	SOCIAL NETWORK ANALYSIS	3	0	0	2	3
4	22B12CS420	SOFTWARE CONSTRUCTION USING KUBERNETICS AND MICRO-SERVICES	3	0	0	3	3

Detailed Syllabus of B.Tech. in Computer Science and Engineering, 2025 Onwards

Approved in Academic Council held on 30 July 2025

Software Development Fundamentals – I

COURSE CODE:XXXXXXXX

COURSE CREDITS: 4

CORE/ELECTIVE: CORE

L-T-P: 3-1-0

Pre-requisite: NONE

Course Objectives:

1. To introduce the software development life cycle and develop problem-solving skills using flowcharts and pseudocode.
2. To provide a strong foundation in C programming concepts including data types, operators, and control flow statements.
3. To enable students to implement and manipulate arrays and pointers for solving computational problems.
4. To develop the ability to write modular programs using functions with various parameter passing techniques.
5. To familiarize students with structures and unions for handling complex data in C.
6. To equip students with skills in file handling to manage text and binary data through C programs.

Course Outcomes:

S.No.	**Course Outcomes**	Level of Attainment
CO-1	Explain the logic for solving problems considering various phases of software development life cycle and depicting them using algorithms and flowcharts	Understand
CO-2	Explain basics of C programming concepts to make decision for solving problems	Understand
CO-3	Demonstrate and contrast different methods for writing modular programs in C	Understand
CO-4	Use various C programming constructs to implement iteration, and recursion	Apply
CO-5	Apply and implement arrays, pointers, structures and file handling for solving real-world problems	Apply

Course Contents:

Unit	Contents	Lectures required
1	Introduction: Introduction to Software Development Life Cycle, Step by step solution to simple problems, developing logic/flow- chart/pseudo code to solve problems like 2D screen saver, simple/logical games, puzzles	6
2	Data types, operators, and Control Flow: Data, variables and constants, data types, operators – binary, unary, ternary, operator precedence, operations using different operators, if, if-else, while, do-while, for, switch-case in C Programming	8
3	Array: Fundamentals of Array, Implementation of 1D/2D Array and related operations like insertion, traversal, updation, etc. in C programming using different problems	7
4	Pointers: Pointers in C, Dynamic memory allocation for 1D/2D array, Arithmetical operations on pointers	5
5	Functions: Introduction to Functions and its implementation in C programming language, Functions using Pass by value, functions using pass by reference, recursive functions	5
6	Structures and Union: Introduction and implementation of Structures and Union in C programming, Array of Structures, Pointer to Structures and related operations like insertion, traversal, updation, etc. in C programming using different problems, Structures using function	5
7	File Handling: Introduction to File, creation of files in C programming language, Modes of File Handling like read, write, update; different types of files like binary file and text file and respective operations like, opening, closing, reading, writing, end of file, traversing the file, for structured and unstructured data	6
Total lectures		42

Suggested Text Book(s):

1. Paul Deitel and Harvey Deitel, “C How to Program”, 9th Edition, Pearson Education, 2022, ISBN: 978-0-13-739839-3
2. E Balagurusamy, “Computing Fundamentals & C Programming”, 2nd Edition, McGraw Hill Education, 2017, ISBN: 978-9352604166
3. Greg Perry and Dean Miller, “C Programming Absolute Beginner's Guide”, 3rd Edition, Que Publishing, 2013, ISBN: 978-0789751980
4. DavidGriffiths and Dawn Griffiths, “Head First C: A Brain-Friendly Guide”, O’Reilly Media, Inc., 2012, ISBN: 978-1449399917

Suggested Reference Book(s):

1. Herbert Schildt, “The Complete Reference C”, 4th Edition, McGraw Hill Education, 2017, ISBN: 978-0070411838
2. Brian W. Kernighan and DennisRitchie, “The C Programming Language”, 2nd Edition, Pearson Education India, 2015, ISBN: 978-9332549449

3. Behrouz A. Forouzan, Richard F. Gilberg, “Computer Science: A Structured Programming Approach Using C”, 3rd Edition, Cengage Learning, 2007, ISBN: 978-8131503638

Other useful resource(s):

1. Link to NPTEL course contents:https://archive.nptel.ac.in/courses/106/105/106105171/
2. Link to topics related to course:
   a. https://www.learn-c.org/
   b. https://www.programiz.com/c-programming
   c. https://www.codechef.com/ide

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 ProgrammeOutcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	2	2	2	1	1	2	2	2	2	**2.0**
CO-2	3	3	2	2	2	1	1	1	2	1	2	**1.8**
CO-3	3	3	3	2	3	1	1	1	2	2	1	**2.0**
CO-4	3	3	3	3	3	1	1	1	2	2	2	**2.3**
CO-5	3	3	3	3	3	1	1	1	1	1	2	**2.2**
Average	**3**	**3**	**2.6**	**2.4**	**2.6**	**1.0**	**1.0**	**1.2**	**1.8**	**1.6**	**1.8**	

Software Development Fundamentals Lab-I

COURSE CODE:XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: NONE

Course Objectives:

1. To develop logic using data types, expressions, and control structures.
2. To implement programs using arrays, strings, and pointers.
3. To apply functions and recursion for modular programming.
4. To use structures and unions for handling complex data.
5. To build menu-driven programs for basic file operations.

Course Outcomes:

S.No.	**Course Outcomes**	Level of Attainment
CO1	Develop programs/logic for data types, expressions and conditionalstructure	Apply
CO2	Perform programs for arrays, strings and pointers	Apply
CO3	Perform programs of functions and recursive functions.	Apply
CO4	Implement programs for structure and union.	Apply
CO5	Implement menu driven programs to perform basic file operations.	Apply

List of Experiments

S.No.	Description	Hours
1	Flow chart and Logic Building: Developing logic/flow-chart/pseudo code to solve problems, simple/logical games, puzzles	4
2	Data Type, Statements, Expressions, Operators: Data, variables and constants, data types, operators – binary, unary, ternary, operator precedence, associativity	2
3	Control Flow: Develop C programs using conditional structure (if, ifelse, nested if), and iterative control structure (do-while, while, for). Implement switch case statement.	4

4	Array and String: Array initialization, reading and writing operations with array, one dimensional, two-dimensional array, strings, and related operations like addition, multiplication, traversal, transpose etc.	4
5	Pointers: Pointers in C, Dynamic memory allocation for 1D/2Darray, Arithmetical operations on pointers, recursivefunctions like palindrome, factorial, fibonacci series,number system etc	4
6	Functions: User defined functions and inbuilt functions, Functions definition, declaration, calling, Pass by value, functions using pass by reference, functions with array	2
7	Structures and Union: Struct keyword, Structure and Union, Structure variable, dot operator, pointer to structures, arrow operator, Array of Structures, structure using functions.	4
8	File Handling: File creation, Modes of File Handling like read, write,update; different types of files like binary file and textfile and respective operations like, opening, closing,reading, writing, end of file, traversing the file forstructured and unstructured data	4
Total Lab hours		28

Suggested/Resources:

1. Paul Deitel and Harvey Deitel, “C HOW TO PROGRAM”, 9th Edition, Pearson Education, 2023, ISBN 978-0-13-739839-3.
2. H. Cooper and H. Mullish, Jaico Publishing House. “Spirit of C”, 4th Edition, Jaico Publishing House,2006.
3. Herbert Schildt. “The Complete Reference C ”, 4th Edition, TMH, 2000.
4. Brian W. Kernighan and Dennis M. Ritchie,“The C Programming Language”, 2nd Edition, PrenticeHall India, New Delhi, 2002.
5. Peter Norton, “Introduction to Computers”, 5th edition, Tata McGraw-Hill, Delhi., 2005.
6. Balaguruswamy, Programming in ANCI C”, 2nd Edition, TMH, 2001.
7. Ashok N. Kamthane , “Programming with ANSI and Turbo C”, Pearson Education, Delhi, 2003.
8. Rajaraman V., “Fundamentals of Computer”, 3rd Edition, Prentice-Hall India, New Delhi, 2005.
9. B. A. Forouzan, R. F. Gilberg “Computer Science: A Structured Programming Approach Using C”,2nd Edition, Thomson Press, New Delhi, 2006.
10. AviSilberschatz, Henry F. Korth, and S. Sudarshan, “Database System Concepts”, 6th edition,McGraw-Hill, 2010.

EvaluationScheme:

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 ProgrammeOutcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	Average
CO-1	3	3	2	2	3	1	1	2	2	2	2	**2.1**
CO-2	3	3	2	2	3	1	1	2	2	2	2	**2.1**
CO-3	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-4	3	3	3	2	3	1	1	2	2	2	2	**2.2**
CO-5	3	3	3	3	3	2	2	3	3	3	3	**2.8**
Average	**3.0**	**3.0**	**2.6**	**2.2**	**3**	**1.2**	**1.2**	**2.2**	**2.2**	**2.2**	**2.2**	

Software Development Fundamentals – II

COURSE CODE:XXXXXXXX

COURSE CREDITS: 4

CORE/ELECTIVE: CORE

L-T-P: 3-1-0

Pre-requisite: C Programming

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 various object-oriented concepts like class and objects, constructors, abstraction, data hiding (access specifiers) etc.	Understand
CO-2	Apply object-oriented concepts like inheritance, polymorphism and templates to various real-world problems.	Apply
CO-3	Apply concepts of data structures such as arrays, stacks, queues, and linked lists to solve real-world problems.	Apply
CO-4	Analyze the source code for possible outcomes, and exceptions, and debugging the errors.	Analyze
CO-5	Design and implement class diagram for varied real life problems using OOPs	Create

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Object Oriented Programming: Comparison of Procedural and Object-Oriented Approach, Characteristics of Object-Oriented Languages, Separation of behaviour and implementation	1
2	OO Concepts using C++: Objects, Classes, Internal representations of Objects, Constructors, Destructors Function and Operator Overloading, Static and Friend Functions	8
3	Inheritance using C++: Base Class, Derived class, Method Overriding, Private and Public Inheritance, Multiple Inheritance.	3

4	Polymorphism using C++: Virtual Functions, Pure Virtual Functions, Abstract Classes, Dynamic Dispatch, Internal representations of method tables, RTTI	3
5	UML/Relationship Implementation in C++: Models, Views and Model Elements, Class Diagram, Relationships of Association, Aggregation, Composition, and Inheritance, etc. and their implementing	6
6	Exceptions, Templates, and STL in C++: Exceptions, Try, Catch and Throw, Re-throwing exceptions, Exception and Inheritance, Function Templates, Overloading Functions Template, Class Templates, Collection classes and iteration protocols (STL)	9
7	Introduction of Data Structures: Fundamentals of data structures, dynamic memory allocation, overview of linear and non-linear DS; Limitations of array and need of linked list, different types of linked list; Stack and Queue, stack and queue operations using STL; Recursion, recursion removal using stack, recursive approaches to solve problems like Tower of Hanoi, N-queen, rat in a maze, etc.	12
Total lectures		42

Suggested Text Book(s):

1. Herbert Schildt, C++: The Complete Reference, McGraw-Hill Osborne Media, 4th Edition, 2017
2. Ellis Horowitz,SartajSahni and Dinesh P. Mehta, Fundamentals of Data Structures in C++, Galgotia Press, 2009
3. Stroustrup B., The C++ Programming Language, Addison Wesley, 4th Edition, 2013
4. John R. Hubbard, Data Structures with C++, Schaum's Outline Series, McGraw Hill, First Edition, 2017.

Suggested Reference Book(s):

1. Robert Lafore, Object Oriented Programming in C++, SAMS, 4th Edition, 2002
2. John Hubbard, Schaum's Outline of Programming with C++, McGraw-Hill, 2nd Edition, 2000

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/106105234
2. Link to NPTEL course contents: https://nptel.ac.in/courses/106102064
3. Link to topics related to course:
   a. https://www.tutorialspoint.com/cplusplus

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 ProgrammeOutcomes(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**	

Software Development Fundamentals Lab-II

COURSE CODE:XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: C Programming

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
CO1	Make use of the concepts related to objects, classes, constructor,destructor, and friend function for solving real-world problems.	Apply
CO2	Apply the principles of encapsulation, inheritance, polymorphismand abstraction in different programming problems.	Apply
CO3	Utilize the Standard Template Library to optimize the object-oriented programming solutions	Apply
CO4	Develop solutions using exception handling for programmingproblems.	Apply
CO5	Develop linear data structures and recursive algorithms using dynamic memory and Standard Template Library (STL).	Apply

List of Experiments

S.No	Description	Hours
1	Output based C++ programs to implement the concepts of Objects, Classes, Internal representations of Objects, encapsulation, Constructors, Destructors, Function and Operator Overloading, Static and Friend Functions	6
2	Write programs in C++ to implement Inheritance, Method Overriding, Private and Public Inheritance, Multiple Inheritance	4
3	Write programs in C++ involving Virtual Functions, Abstract Classes, Dynamic Dispatch, operator overriding, etc.	2

4	Write programs in C++ to implement the relationships of Association, Aggregation, Composition, and Inheritance	4
5	Write programs in C++ to handle Exceptions, Re-throwing exceptions, Function Templates, Overloading Functions Template, Class Templates, Collection classes and iteration protocols (STL)	4
6	Programs for Dynamic Memory Allocation, Abstract Data Types; Templates and STL to implementation linear data structures like Array, Linked List, Stack and Queue; implementation of recursive algorithm to solve different problems	8
Total Lab hours		28

Suggested/Resources:

1. Herbert Schildt, C++: The Complete Reference, McGraw-Hill Osborne Media, 4th Edition, 2017.
2. RamezElmasri, Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition, 2016
3. Walter Savitch, Kenrick Mock," Absolute C++", Pearson, 6th Edition, 2016.
4. E BALAGURUSAMY, Object Oriented Programming with C++, McGraw-Hill Education (India), 8th Edition, 2020.
5. AviSilberschatz, Henry F. Korth, and S. Sudarshan, "Database System Concepts", McGraw-Hill, 7thedition, 2019.
6. Cay S. Horstmann, Big C++: Late Objects, Wiley, 3rd edition, 2017.
7. Stroustrup B., The C++ Programming Language, Addison Wesley, 4th Edition, 2015.
8. Robert Lafore, Object Oriented Programming in C++, SAMS, 4th Edition, 2002.
9. John Hubbard, Schaum's Outline of Programming with C++, McGraw-Hill, 2nd Edition, 2000.

EvaluationScheme:

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 ProgrammeOutcomes (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**	

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	Remember, 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 grammarsand 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 toTuring machines, numerical functions applied to Turing machines, concept of halting problem. Grammars, properties ofrecursive languages, concept of polynomial decidable	Evaluate

Course Contents:

Unit	Contents	Lectures required
1	General introduction, Complexity theory, Computability theory, Automata theory, Mathematical preliminariesInfinite 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 RecursivelyEnumerable 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/

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 ProgrammeOutcomes(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**	

Data Structures

COURSE CODE:XXXXXXXX

COURSE CREDITS: 4

CORE/ELECTIVE: CORE

L-T-P: 3-1-0

Pre-requisite: Programming languages C and C++

Course Objectives:

1. To understand and implement linear data structures and basic representations of trees and graphs.
2. To apply various searching and sorting algorithms with analysis of their time and space complexity.
3. To implement efficient hashing techniques for optimized data access.
4. To design and apply advanced tree and heap data structures for effective data management
5. To use graphs, special trees, and string data structures in solving complex computational problems.

Course Outcomes:

S.No.	**Course Outcomes**	Level of Attainment
CO-1	Understand and implement linear data structures and basic tree/graph representations.	Remember, Understand
CO-2	Apply searching and sorting algorithms with time and space complexity analysis.	Evaluate
CO-3	Implement hashing techniques and evaluate their efficiency.	Evaluate
CO-4	Design and apply advanced tree and heap structures for optimized data organization and retrieval.	Apply
CO-5	Use graphs, special trees, and string data structures in solving computational problems.	Apply

Course Contents:

Unit	Contents	Lectures required
1	Overview of linear data structures: array, linked list, stack, and queue	3
2	Searching and sorting algorithms – Interpolation Search, Median Search; Hashing – Hash Table, Chaining, Probing; Sorting – Merge, Quick, Radix, Bucket, and Count; Time and Space complexity analysis of searching and sorting algorithms	7
3	Implementation of Multi List, Binary Tree, K-ary Tree, Tree traversals, Threaded Binary Tree, Priority Queue using Binary Heap, Binomial Heap, and Fibonacci Heap, Heap Sort	10

4	Binary Search Tree (BST), Balanced BST: AVL Tree and RB Tree; Multiway Tree: B Tree and B+ Tree	10
5	Fundamentals of Graph, Adjacency Matrix and List; Graph Traversal using DFS and BFS, Basic Algorithms – Shortest Path and Minimum Spanning Tree	6
6	Interval Tree, Segment Tree, and String Data Structures: Suffix Tree, Tries, Suffix Array	6
Total Lectures		42

Suggested Text Book(s):

1. Aaron M. Tanenbaum, *Data Structures Using C*, Pearson Education, Year not specified.
2. Ellis Horowitz and Sartaj Sahni, *Fundamentals of Data Structures*, Illustrated Edition, Computer Science Press, Year not specified.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 3rd Edition, MIT Press, 2009.
4. Dinesh P. Mehta and Sartaj Sahni, *Handbook of Data Structures and Applications*, 2nd Edition, Chapman and Hall/CRC Computer and Information Science Series, CRC Press, Year not specified.

Suggested Reference Book(s):

1. Mark Allen Weiss, *Algorithms, Data Structures, and Problem Solving with C++*, Illustrated Edition, Addison-Wesley Publishing Company, Year not specified.
2. Ellis Horowitz, Sartaj Sahni, and Dinesh P. Mehta, *Fundamentals of Data Structures in C++*, Galgotia Press, 2009.
3. Seymour Lipschutz, *Data Structures with C*, Schaum's Outline Series, McGraw Hill, 2010.
4. Sara Baase and Allen Van Gelder, *Computer Algorithms: Introduction to Design and Analysis*, Pearson Education, Year not specified.
5. Thomas H. Cormen, *Algorithms Unlocked*, MIT Press, Year not specified.
6. DebasisSamanta, *Classic Data Structures*, Prentice Hall of India, Year not specified.

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

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	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	2	2	2	3	1	1	1	1	1	2	**1.7**
CO-4	3	3	3	2	3	1	1	1	1	1	2	**1.9**
CO-5	3	3	3	2	3	1	1	1	1	1	2	**1.9**
Average	**3**	**2.6**	**2.4**	**1.8**	**2.8**	**1**	**1**	**1**	**1**	**1**	**2**	

Data StructuresLab

COURSE CODE:XXXXXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: C/C++

Course Objectives:

1. To understand and implement linear data structures such as arrays, linked lists, stacks, queues, and sparse matrix representations.
2. To learn and apply various searching, sorting, and hashing algorithms along with their complexity analysis.
3. To implement different types of trees including binary trees, K-ary trees, threaded trees, and various heap structures.
4. To design and apply balanced and multiway trees like AVL, Red-Black, B, and B+ Trees for efficient data handling.
5. To develop graph traversal algorithms and advanced structures such as Tries, Segment Trees, and Interval Trees for solving complex computational problems.

Course Outcomes:

S.No.	**Course Outcomes**	Level of Attainment
CO1	Implement linear data structures such as arrays, linked lists, stacks, queues, and multi-linked lists for sparse matrix representation.	Remember, Understand
CO2	Apply searching, sorting, and hashing techniques including interpolation search, merge sort, radix sort, and hash table operations.	Evaluate
CO3	Implement various tree structures including binary trees, K-ary trees, threaded trees, and heaps like binary and Fibonacci heaps.	Apply
CO4	Develop balanced and multiway trees such as AVL, Red-Black, B, and B+ Trees for efficient data organization and access.	Apply
CO5	Implement graph traversal algorithms (DFS &BFS), shortest path, minimum spanning tree, and advanced structures like Tries, Segment and Interval Trees.	Apply

List of Experiments

S.No	Description	Hours
1	Implementation of linear data structures like, array, linked lists, stack, queue and related operations, multi linked list for sparse matrix representation	4
2	Implementation of searching and sorting algorithms like, linear search, binary Search, interpolation search, median Search; Hashing – Hash Table, Chaining, Probing; Sorting – Merge, Quick, Radix, Bucket, and Count	4
3	Implementation of Binary Tree, K-ary Tree, Threaded Tree, Priority Queue using Binary Heap and Fibonacci Heap, Heap Sort	6
4	Implementation of Binary Search Tree (BST), Balanced BST: AVL Tree & RB Tree; Multiway Tree: B Tree & B+ Tree	6
5	Writing programs to traverse graph using DFS & BFS and implement the algorithms for Shortest Path and Minimum Spanning Tree	4
6	Implementation of Interval Tree, Segment Tree, and Tries	4
Total Lab hours		28

Suggested/Resources:

1. Ellis Horowitz, Sartaj Sahni and Dinesh P. Mehta, Fundamentals of Data Structures in C++, Galgotia Publications, 2009.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithms, 4th Edition, MIT Press, 2022.
3. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 4th Edition, Pearson Education, 2014.
4. Dinesh P. Mehta and Sartaj Sahni, Handbook of Data Structures and Applications, 2nd Edition, Chapman & Hall/CRC, 2018.
5. Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Addison-Wesley, 2011.
6. Jon Kleinberg and ÉvaTardos, Algorithm Design, Pearson Education, 2006.
7. Link to topics related to course:
   i. https://www.geeksforgeeks.org
   ii. https://www.coursera.org/specializations/data-structures-algorithms

EvaluationScheme:

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	1	1	2	**1.6**
CO-2	3	3	2	2	3	1	1	1	1	1	2	**1.8**
CO-3	3	2	2	2	3	1	1	1	1	1	2	**1.7**
CO-4	3	3	3	2	3	1	1	1	1	1	2	**1.9**
CO-5	3	3	3	2	3	1	1	1	1	1	2	**1.9**
Average	**3**	**2.6**	**2.4**	**1.8**	**3**	**1**	**1**	**1**	**1**	**1**	**2**	

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.	Remember, 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	Remember, 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. Link to NPTELCourse Contents:
   https://archive.nptel.ac.in/courses/106/105/106105175/

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 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	**1.73**
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, groupfunctions, count functions etc.	2
4	To practice group by, having clause and special operators such as between, like, in etc.	2

5	To practice 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 and Functions	6
10	To implement Cursors and Triggers	2
11	To implement Transaction Processing Commands: Commit, Rollback, Save Point and Time Stamps	2
12	To implement Data Control Language Commands: Grant and Revoke	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.

EvaluationScheme:

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:XXXXXX

COURSECREDIT: 2

CORE/ELECTIVE: CORE

L-T-P: 1-0-2

Pre-requisite: Programming Languages C, C++

Course Objective:

1. Introduces basic understanding of UNIX OS, UNIX commands and File system.
2. To familiarize students with the Linux environment.
3. To make student learn fundamentals of shell script and shell programming.
4. Emphases are on making student familiar with UNIX environment and issues related to it.

CourseOutcomes:

S.No.	**Course Outcomes**	Attainment Level
CO1	Able to run various UNIX commands on a standard UNIX/LINUX Operating system	Remember, Understand
CO2	Able torunC/ C++programs on UNIX	Evaluate
CO3	Abletodoshellprogrammingon UNIXOS.	Apply
CO4	AbletounderstandandhandleUNIXsystem calls	Apply

Listof Experiments:

S. No.	Description	(L+P) Hours
1	Introduction to Unix Operating System and overview to Open-Source Software, Writing C program in Unix environment using GCC compiler	1+2
2	working with vi editor, vi editor and basic UNIX commands: Creating and editing a text file with the vi text editor using the standard vi editor commands	2+4
3	The UNIX file system, working with directories, input-output redirection, Pipes, Processes	2+4
4	UNIX Shell Programming: Working with filters: grep, sed and awk, pick and news command, get and put customizing the cal command, while and until loops, traps, zap	4+8

5	Programming with standard I/O	3+6
6	UNIX System Calls: low level I/O, File system- directories, inodes, process, signals and interrupts	2+4
	Total Hours	14+28

Suggested Textbook(s):

1. Brian W. Kernighan and Rob Pike, *The UNIX Programming Environment*, Prentice Hall India. (Available in the LRC and as an eBook on the student resource portal)
2. Sumitabha Das, *UNIX: Concepts and Applications*, Tata McGraw Hill, Latest Edition.

Suggested Reference Book(s):

1. Sumitabha Das, *UNIX Concepts and Applications*, McGraw Hill Education (India) Pvt. Ltd.
2. Stephen Prata, *Advanced UNIX: A Programmer's Guide*, The Wait Group, Sams Publishing.
3. Jack Tackett Jr. and David Gunter, *Using Linux: The Most Complete Reference*, 3rd Edition, Que Publishing.

Evaluation Scheme:

S. No	Exam	Marks
1	Mid Sem. Evaluation (P-1)	20
2	Internal Assessment (May include attendance, class participation, lab file records, experiment execution, viva voce, quizzes, mini-projects etc.)	40
3.	Written Theory Exam	40
	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	1	1.6
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
Average	2.67	2.5	2.5	2.25	2.75	2.25	1.5	1.5	1.5	1.25	2	

Object Oriented 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 (COs)**	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.	Remember, 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, Functioning of JVM and JDK, Structure of 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. Course links

   i) NPTEL – Programming in Java (IIT Kharagpur) | https://nptel.ac.in/courses/106105084
   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) TutorialsPoint – Java Programming | https://www.tutorialspoint.com/java/index.htm
   vi) 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
CO1	2	2	3	3	3	1	1	1	1	1	3	**1.90**
CO2	2	2	3	3	3	2	1	2	2	1	3	**2.36**
CO3	2	2	3	3	3	2	1	2	2	1	3	**2.18**
CO4	2	2	3	3	3	2	1	2	2	1	3	**2.18**
CO5	2	2	3	3	3	2	1	1	1	1	3	**2.0**
CO6	2	2	3	3	3	2	1	2	2	1	2	**2.18**
CO7	2	2	3	3	3	1	1	1	1	1	2	**1.82**
Average	**2.3**	**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

EvaluationScheme:

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**	-	-	-	

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

EvaluationScheme:

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.	Remember, Understand
CO-2	Create processes and threads.	Create, Evaluate
CO-3	Develop algorithms for process scheduling for a givenspecification of CPU utilization, Throughput, Turnaround Time,Waiting Time, and Response Time.	Create, Evaluate
CO-4	For a given specification of memory organization develop thetechniques 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/Omanagement functions in OS as part of a uniform deviceabstraction by performing operations for synchronizationbetween 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, ResponseTime; Scheduling Algorithms: Pre-emptive and Non preemptive,FCFS, SJF, Priority, Round Robin, Multiple processor scheduling	7
4	Inter-process Communication: Critical Section, RaceConditions, Mutual Exclusion, Hardware Solution, StrictAlternation, Peterson's Solution, Lamport's BakeryAlgorithm, The Producer\ Consumer Problem, Semaphores,Classical IPCProblems: Reader's & Writer Problem, Dinning PhilosopherProblem etc. Monitors	10
5	Deadlocks: Definition, Necessary and sufficient conditionsfor Deadlock, Deadlock Prevention, Deadlock Avoidance:Banker's algorithm, Deadlock detection and Recovery.	4
6	Memory Management: Basic concept, Logical and Physicaladdress map, Memory allocation: Contiguous Memoryallocation – Fixed and variable partition–Internal andExternal fragmentation and Compaction; Paging: Principle ofoperation – Page allocation – Hardware support for paging,Protection and sharing, Disadvantages of paging.	6
7	Virtual Memory: Basics of Virtual Memory – Hardware andcontrol structures – Locality of reference, Page fault,Working Set, Dirty page/Dirty bit – Demand paging, PageReplacement algorithms: Optimal, First in First Out (FIFO),Not recently used (NRU) and LeastRecently 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

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 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,printingEnvironment,customizing networking,x-applicationsunix 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

EvaluationScheme:

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.	Remember, 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

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 ProgrammeOutcomes(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:XXXXX

COURSE CREDITS: 1

CORE/ELECTIVE: CORE

L-T-P: 0-0-2

Pre-requisite: C Language

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
CO1	To understand the working of NS2 and other network simulators.	Apply
CO2	To implement the concepts of error control and flow control in data link layer.	Create, Evaluate
CO3	To implement routing algorithms, broadcasting, encryption and decryption, ad congestion control in network layer.	Create, Evaluate
CO4	To analyze packets dropped, data rate, throughput, congestion, and network performance	Evaluate

List of Experiments

S.No.	Description	Hours
1	Overview of NS2 and other network simulators, Hands-on on NS2 simulator	6
2	Write programs to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP	2
3	Implement the data link layer framing methods,	2
4	Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism	2
5	Implement Dijsktra's algorithm to compute the shortest path through a network	2

6	Implement distance vector routing algorithm for obtaining routing tables at each node, For given subnet of hosts, obtain the broadcast tree for the subnet	2
7	Implement data encryption and data decryption	2
8	Write programs for congestion control and frame sorting technique used in buffers	2
9	Use NS2 Simulator to perform following:Find the Number of Packets Dropped, Find the Number of Packets Dropped by TCP/UDP	2
10	Find the Number of Packets Dropped due to Congestion	2
11	Compare Data Rate & Throughput, Plot Congestion for Different Source/Destination	2
12	Determine the Performance with respect to Transmission of Packets	2
Total Lab hours		28

Suggested/Resources:

1. James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet" 3rd Edition Pearson Education
2. Andrew S. Tanenbaum, "Computer Networks" 4th Edition PHI.
3. UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998, ISBN 0-13-490012-XSunilaGollapudi, "Practical Machine Learning", Packt Publishing Ltd
4. Arnold Robbins, "UNIX in a Nutshell", O"Reilly 4th EditionWilliRichert, "Building Machine Learning Systems with Python", Packt Publishing Ltd.
5. David I. Schawartz, "Introduction to UNIX", Prentice Hall, Second Edition
6. BEHROUZ a. Forouzan and Richard F. Gilberg, "UNIX and Shell Programming: A Textbook"
7. NS Simulator for Beginner"s, Lecture notes Univ. de Los Andes, France.
8. Link 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

EvaluationScheme:

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 ProgrammeOutcomes (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	2	**2.6**
CO-4	3	3	3	2	2	3	2	2	3	3	2	**2.5**
Average	**3**	**3**	**3**	**2**	**2.5**	**2.5**	**2.2**	**2**	**2.5**	**3**	**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.

EvaluationScheme:

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 ProgrammeOutcomes (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 Path in a Matrix, Egg Dropping Problem, Diameter of Tree, Maximum Path Sum, Wildcard Matching, DP on subsets, strings, grids,	10
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

EvaluationScheme:

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**	**-**	**-**	**-**	