

Detailed Syllabus of B.Tech. in Artificial
Intelligence & Machine Learning, 2026 Onwards
(Only for 1st and 2nd Semester)

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. David Griffiths 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 Dennis Ritchie, "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>

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	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: XXXXXXXXX

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 conditional structure	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	4

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.	2
5	Pointers: Pointers in C, Dynamic memory allocation for 1D/2D array, Arithmetical operations on pointers, recursive functions 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 for structured and unstructured data	4
9	Project	
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.

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	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, Sartaj Sahni 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, polymorphism and 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 programming problems.	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	6

	and Friend Functions	
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, 7th edition, 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 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	