PART A: Introduction				
Program: **Under Graduate**		Class: **B.Sc.**	Year: **First Year**	Session: **2025-26**
Subject: **Computer Science**				
1.	Course Code			
2.	Course Title	C-2(TH): **Programming Methodologies & Data Structures (Using C/C++)**		
3.	Course Type (Core Course/Elective/Generic Elective/ Vocational	**Core Course**		
4.	Pre-Requisite (if any)	To study this course, Mathematics of 12th standard is desirable.		
5.	Course Learning Outcomes (CLO)	On completion of this course, learners will be able to: 1. Develop *(Level-6)* simple algorithms and flow charts to solve a problem with programming using top-down design principles; 2. Writing *(Level-6)* efficient and well-structured computer algorithms/programs; 3. Formulate *(Level-6)* iterative solutions and array processing algorithms for problems; 4. Use *(Level-3)* recursive techniques, pointers and searching methods in programming; 5. Implement *(Level-3)* fundamental data structures & accustomed to the description of algorithms in both functional and procedural styles; 6. Understand *(Level-2) the* complexity of basic operations like insert, delete, search on these data structures. 7. Select appropriate *(Level-5)* data structure ~~to~~ suitable to different models; 8. Design *(Level-6)* programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc.; 9. Assess *(Level-5)* efficiency tradeoffs among different data structure implementations; 10. Implement *(Level-3)* and know the applications of algorithms for searching and sorting etc.; 11. Identify *(Level-2)* the contributions of Indians in the field of programming and data structures. ***Note:*** *Level of Bloom's Taxonomy is mentioned in the brackets.*		
6.	Credit Value	Theory - **4 Credits**		
7.	Total Marks	Max. Marks: **30+70**	Min. Passing Marks: **35**	

PART B: **Content of the Course**		
No. of Lectures (in hours per week): **04 Hrs. per week**		
Total No. of Lectures: **60** Hrs.		
Module	**Topics**	**No. of Lectures**
I	*Relevant Indian Knowledge System (IKS) Inclusions:* Algorithmic thinking in Ancient India. The Panini Grammar System (Ashtadhyayi). The Chandas Shastra (Sanskrit Prosody) a recursive structure, The Brahmagupta Algorithm (7th century CE) an introduction to zero and place value notation. *Data Structures & Computational Methods in Ancient India:* Vedic method of data structuring – preservation of ancient manuscripts (e.g., Rig Veda) using hierarchical structures. Resemblance of Graph Theory with the Indian Temple Architecture (the connectivity principles of temple design and city planning). Resemblance of efficient Sorting & Searching techniques with Ancient Indian classification methods in Ayurveda & Sanskrit texts. The Buddhist Numerical Sorting Method (Bhāskara II). *Introduction to Programming:* Program Concept, Characteristics of Programming, Stages in Program Development, Algorithms, Notations, Design, Flowcharts, Types of Programming Methodologies. *Introduction to C/C++ Programming:* Basic Program Structure in *C/C++*, Data Types, Variables, Constants, Operators and Basic I/O. *Variables:* Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (*getc, getchar, putc, putchar* etc.), Formatted and Console I/O [*printf(), scanf(), cin(), cout()*], Using Basic Header Files (*stdio.h, iostream.h, conio.h* etc.), Simple Expressions in *C/C++* (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions. ***Suggested activities for experiential learning:*** 1. *Comparative Analysis: Research how Panini's grammar rules resemble formal grammar in programming languages.* 2. *Algorithm Simulation: Implement Brahmagupta's place-value system using C/C++.* 3. *Keyword Identification Exercise: Analyze the similarity between Sanskrit syntax and C/C++ keywords (e.g., structure in Sanskrit grammar vs. C/C++ struct).* 4. *Basic I/O Project: Implement a console-based quiz using formatted I/O.*	10

II	Conditional Statements *if* construct, *switch-case* construct. *Iterative Statements: while, do-while,* and *for* loops, use of *break* and *continue* in loops, Using Nested Statements (Conditional as well as Iterative). *Functions:* Top-Down Design, Pre-defined Functions, Programmer defined Functions. Local Variables and Global variables, Functions with Default Arguments, Call-By-Value and Call-By-Reference, Parameters, Recursion. *Introduction to Arrays:* Declaration and Referring Arrays, Arrays in Memory, Initializing Arrays. Arrays in Functions, Multi-Dimensional Arrays. ***Suggested activities for experiential learning:*** 1. *Code Debugging Challenge: Assign buggy C/C++ programs for students to debug and improve.* 2. *Concept Visualization: Use flowcharts and pseudocode tools to map variable types and memory usage.* 3. *Conditional Logic Game: Design a decision-based game using **if-else** and **switch-case**.* 4. *Function Optimization Task: Analyze and optimize recursive vs. iterative function performance.* 5. *Nested Loops Visualization: Represent nested loops using Pascal's Triangle visualization.* 6. *Real-World Decision-Making Simulation: Create a banking/ATM system that demonstrates nested loops and conditional logic.*	08
III	*Structures:* Member Accessing, Pointers to Structures, Structures and Functions, Arrays of Structures. *Unions:* Declaration and Initialization. *Strings:* Reading and Writing Strings, Arrays of Strings, String and Function, Strings and Structure, Standard String Library Functions. *Searching Algorithms:* Linear Search, Binary Search. *File Handling:* Use of files for data input and output, merging and copying files. Ayurvedic Classification System: Map hierarchical classification of medicinal plants to data structures like arrays and pointers, Shulba Sutras for Spatial Computations: Understanding recursive patterns in Shulba Sutras and their application in functions. ***Suggested activities for experiential learning:*** 1. *Mapping Ayurvedic Taxonomy to Data Structures: Represent Ayurvedic classification of herbs using arrays and nested data structures.* 2. *Function Optimization Project: Implement recursive and iterative functions to compare execution time.* 3. *Group seminar and Online quiz based on searching algorithm and file handling*	08

	4. *Vedic Sorting Implementation: Develop a sorting algorithm inspired by Ayurvedic classification techniques.*	
IV	*Data Structure:* Basic concepts, Linear and Non-Linear data structures. *Algorithm Specifications:* Introduction, Recursive algorithms, Data Abstraction, Performance analysis. *Linked List:* Singly Linked Lists, Operations, Concatenating, circularly linked lists-Operations for Circularly linked lists, Doubly Linked Lists- Operations. *Array:* Representation of single, two-dimensional arrays, sparse matrices-array and linked representations. *Stack:* Operations, Array and Linked Implementations, Applications Infix to Postfix Conversion, Postfix Expression Evaluation, Recursion Implementation. *Linked List Concept in Indian Knowledge:* Ancient Sanskrit texts used linked hierarchical structuring for preserving information (e.g., Vedic oral tradition). *Stack Analogy in Nyaya Logic:* Indian logical frameworks used last-in-first-out (LIFO) reasoning similar to stack operations. ***Suggested activities for experiential learning:*** 1. *Chart Preparation: Prepare charts Linked List, Array & Stack* 2. *Stack Simulation Exercise via Role-Play: Implement a LIFO-based task scheduler.* 3. *Linked List Research Assignment: Compare linked list pointer-based structure with ancient manuscript referencing.* 4. *Students will map historical Guru-Shishya Parampara in the form of a singly linked list (E.g., Vyasa → Shuka → Gaudapada → Govindapada → Adi Shankaracharya)* *Implement this as a linked list in C/C++, where each node represents a teacher and links to their disciples.*	12
V	*Queue:* Definition, Operations, Circular Queue-Insertion and Deletion Operations, Dequeue (Double Ended Queue), Priority Queue-Implementation. *Trees:* Representation of Trees, Binary tree, Properties of Binary Trees, Binary Tree Representations- Array and Linked Representations, Binary Tree Traversals, Threaded Binary Trees. *Heap:* Definition, Insertion, Deletion. *Buddhist Numerical Sorting:* Bhaskara II's early classification techniques, *Efficient Searching in Ayurveda:* Ayurvedic medicinal classification principles resemble hashing and tree-based sorting. ***Suggested activities for experiential learning:*** 1. *Queue-based Ticketing System: Develop a queue system (FIFO) for handling real-world ticket processing.*	10

	2. *Search Algorithm Hackathon:* Implement linear, binary, and hashing techniques to solve real-world problems. 3. *Sorting Race:* Students compete to optimize sorting algorithms based on Ayurvedic classification techniques. 4. *Data Organization Challenge:* Create efficient storage models for Ayurveda medicinal records using tree-based structures. 5. *Comparative Study:* Research how Vedic knowledge management compares with modern database indexing.	
VI	*Graphs:* Graph ADT, Graph Representations, Graph Traversals, Searching. *Hashing:* Introduction, Hash tables, Hash functions, Overflow Handling. *Sorting Methods:* Comparison of Sorting Methods, Search Trees - Binary Search Trees, AVL Trees- Definition and Examples. *Tree Representation in Ancient India:* Genealogy (Gotra System) as an early example of hierarchical tree structures. *Graph Connectivity in Temple Design:* Principles of temple planning akin to graph traversal. ***Suggested activities for experiential learning:*** 1. *Family Tree Implementation:* Use binary trees to model ancient Indian lineage systems. 2. *Graph Problem Solving:* Model Indian temple network connectivity using graph algorithms. 3. *Heap Data Structure Exploration:* Implement heap sorting for priority-based Ayurveda classification. 4. *Shortest Path Challenge:* Use Dijkstra's Algorithm to optimize ancient pilgrimage route planning. 5. *Graph Theory Workshop: Study the resemblance of temple architecture to graph connectivity and model it using Graphviz/NetworkX.* 6. *Tree Traversal Experiment: Implement tree traversal to simulate genealogy in Vedic lineage texts.*	10
VII	Indian Contribution to the field: Innovations in India, origin of Julia Programming Language, Indian Engineers who designed new programming languages, open-source languages, Dr. Sartaj Sahni — Computer Scientist - Pioneer of data structures, Murthy's Early Work in Software Development. Julia Programming Language's Indian Origins. ***Suggested activities for experiential learning:*** 1. *Research Presentation:* Students present on Indian-origin computer scientists. 2. *Coding Tribute:* Implement an algorithm inspired by Sartaj Sahni's data structure optimizations. 3. *Innovation Showcase:* Identify Indian-origin open-source projects and contribute to them. 4. *Documentary Screening & Discussion:* Screen a documentary on India's tech evolution and discuss its impact.	2

PART C: Learning Resources

Textbooks, Reference Books, Other Resources

Suggested Readings:

- Lipschutz: Schaum's outline series Data structures, Tata McGraw-Hill
- Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015
- E. Balguruswamy, "C++ " TMH Publication ISBN 0-07-462038-X
- Herbertz Shield, "C++ The Complete Reference "TMH Publication ISBN 0-07-463880-7
- R. Lafore, "Object Oriented Programming C++"
- N. Dale and C. Weems, Programming and problem solving with C++: brief edition, Jones & Bartlett Learning.
- Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning.
- Tony Gaddis, Starting Out With Python
- Kenneth A. Lambert, Fundamentals of Python
- Kiparsky, P. (2009). Panini as a Formalist.
- Joseph, G. G. (1991). The Crest of the Peacock: Non-European Roots of Mathematics.
- Bhaskara II (12th Century), Lilavati and Bijaganita.
- Staal, F. (2006). The Science of Language and Logic in India.
- Pingree, D. (1978). Mathematical Astronomy in India.
- Kosambi, D. D. (1948). The Culture and Civilization of Ancient India.
- Sahni, S. (2005). Data Structures, Algorithms, and Applications in C++.

Suggested online resources:

http://www.ndl.gov.in/he_document/ekumbh/97
http://www.ndl.gov.in/he_document/nptel/IN_N_1_C_S_a_E_9093_N_P_D_S_a_A_u_P_12265_12266

https://archive.nptel.ac.in/courses/106/105/106105171/
https://archive.nptel.ac.in/courses/106/105/106105234/
https://archive.nptel.ac.in/courses/106/101/106101208/
https://archive.nptel.ac.in/courses/106/106/106106133/
https://epgp.inflibnet.ac.in/Home/ViewSubject?catid=fBYckQKJvP3a/8Vd3L08tQ==
https://nptel.ac.in/courses/106105151

Part D: Assessment and Evaluation

Suggested Continuous Evaluation Methods:
Maximum Marks: **100**
Continuous Comprehensive Evaluation (CCE): **30** Marks
University Exam (UE): **70** Marks

Internal Assessment: Continuous Comprehensive Evaluation (CCE)	Class Test Assignment/Presentation	**Total Marks: 30**
External Assessment: University Exam (UE) Time: 03.00 Hours	Section (A): Objective type Section (B): Short Questions Section (C): Long Questions	**Total Marks: 70**

PART A: Introduction				
Program: **Under Graduate**		Class: **B.Sc.**	Year: **First Year**	Session: **2025-26**
Subject: **Computer Science**				
1.	Course Code			
2.	Course Title	C-2(PR): **Programming Methodologies & Data Structures (Using C/C++) (Lab)**		
3.	Course Type (Core Course/Elective/Generic Elective/ Vocational	Core Course		
4.	Pre-Requisite (if any)	To study this course, Mathematics of 12thstandard is desirable.		
5.	Course Learning Outcomes (CLO)	On completion of this course, learners will be able to: 1. Develop *(Level-6)* simple algorithms and flow charts to solve a problem with programming using top-down design principles. 2. Writing *(Level-6)* efficient and well-structured computer algorithms/programs. 3. Formulate *(Level-6)* iterative solutions and array processing algorithms for problems. 4. Apply *(Level-4)* recursive techniques, pointers and searching methods in programming. 5. Select *(Level-5)* appropriate data structure suitable to different models; 6. Implement *(Level-3)* and know the applications of algorithms for searching and sorting etc.; ***Note:*** *Level of Bloom's Taxonomy is mentioned in the brackets.*		
6.	Credit Value	Practical **-2 Credits**		
7.	Total Marks	Max. Marks: **100**	Min. Passing Marks: **35**	

PART B: **Content of the Course**	
No. of Lab Practical (in hours per week): **2 per week**	
Total No. of Lab.: **60 Hrs.**	
Suggestive list of Practical	No. of Labs.

Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code in **C/C++**, execute and test it. Students should be given assignments on following:

1. a. To learn elementary techniques involving arithmetic operators and mathematical expressions, appropriate use of selection (if, switch, conditional operators) and control structures.

   b. Learn how to use functions and parameter passing in functions, writing recursive programs.
2. Write a program to swap the contents of two variables.
3. Write a program for finding the roots of a Quadratic Equation.
4. Write a program to find area of a circle, rectangle, square using switch case.
5. Write a program to check whether a given number is even or odd.
6. Write a program to print table of any number.
7. Write a program to print Fibonacci series.
8. Write a program to find factorial of a given number.
9. Write a program to convert decimal (integer) number into equivalent binary number.
10. Write a program to check given string is palindrome or not.
11. Write a program to perform multiplications of two matrices.
12. Write a program to print digits of entered number in reverse order.
13. Write a program to print sum of two matrices.
14. Write a program to print multiplication of two matrices.
15. Write a program to generate even/odd series from 1 to 100.
16. Write a program whether a given number is prime or not.
17. Write a program for call by value and call by reference.
18. Write a program to generate a series 1+1/1 !+2/2!+3/3!+----------+n/n!
19. Write a program to create a pyramid structure

    *
    **
    ***
    ****
20. Write a program to create a pyramid structure.

    1
    12
    123
    1234
21. Write a program to check entered number is Armstrong or not.
22. Write a program for traversing an Array.
23. Write a program to input N numbers, add them and find average.
24. Write a program to find largest element from an array.
25. Write a program for Linear search.
26. Write a program for Binary search.
27. Write a program for Bubble sort.
28. Write a program for Selection sort.

PART C: Learning Resources

Textbooks, Reference Books, Other Resources

Suggested Readings:

- Problem Solving and Program Design in C, J. R. Hanly and E. B. Koffman, Pearson, 2015
- E Balguruswamy, "C++ " TMH Publication ISBN 0-07-462038-X
- Herbertz Shield, "C++ The Complete Reference "TMH Publication ISBN 0-07-463880-7 R. Lafore, "Object Oriented Programming C++"
- N. Dale and C. Weems, "Programming and problem solving with C++", brief edition, Jones & Bartlett Learning.
- Adam Drozdek, "Data Structure and Algorithm in C++", Third Edition, Cengage Learning.
- Sartaj Sahani, "Data Structure, Algorithm and Applications with C++", McGraw Hill.
- Robert L. Kruse, Data Structure and Program Design in C++" Pearson.
- D.S. Malik, Data Structure using C++, Second Edition, Cengage Learning.
- M. A. Weiss, Data structures and Algorithm Analysis in C, 2nd edition, Pearson.
- Lipschutz: Schaum's outline series Data structures, Tata McGraw-Hill

Suggestive digital platform web links:

https://www.yout-ube.com/watch?v=BClS40yzssA
https://www.yout-ube.com/watch?v=vLnPwxZdW4Y&vl=en
https ://www.yout-ube.com/watch?v=Umm I ZQ5 ltZw
https://nptel.ac.in/courses/106/ 106/ 106106127/

Suggested equivalent online courses:

https://nptel.ac.in/courses/106/105/ 106105 1 5 1 /
https://nptel.ac.in/courses/106/105/ 106105 1 7 1/

PART D: Assessment and Evaluation

Suggested Continuous Evaluation Methods:

Internal Assessment	Marks	External Assessment	Marks
Class Interaction/Quiz	NIL	Viva Voce on Practical (20 marks)	100
Attendance		Practical Record File (20 marks)	
Assignments (Charts/Model/Seminars / Technology Dissemination/ Excursion/ Lab visit/ Industrial Visit)		Table Work / Exercise Assigned (60 marks)	
	Total Marks: 100		

Prof. Navita Shrivastava
Chairman Board of Studies