

B.Sc- Information Technology Syllabus under CBCS Pattern with effect from 2026-2027 onwards



PERIYAR UNIVERSITY

PERIYAR PALKALAI NAGAR

SALEM-636011

DEGREE OF BACHELOR OF SCIENCE

Syllabus for

B.Sc., INFORMATION TECHNOLOGY

(SEMESTER PATTERN- CBCS)

(For Candidates admitted in the colleges affiliated to

Periyar University from 2026-2027 onwards)

1. Introduction

B.Sc. Information Technology

Education is the key to development of any society. Role of higher education is crucial for securing right kind of employment and also to pursue further studies in best available world class institutes elsewhere within and outside India. Quality education in general and higher education in particular deserves high priority to enable the young and future generation of students to acquire skill, training and knowledge in order to enhance their thinking, creativity, comprehension and application abilities and prepare them to compete, succeed and excel globally. Learning Outcomes-based Curriculum Framework (LOCF) which makes it student-centric, interactive and outcome-oriented with well-defined aims, objectives and goals to achieve. LOCF also aims at ensuring uniform education standard and content delivery across the state which will help the students to ensure similar quality of education irrespective of the institute and location.

Computer Science is the study of quantity, structure, space and change, focusing on problem solving, application development with wider scope of application in science, engineering, technology, social sciences etc. throughout the world in last couple of decades and it has carved out a space for itself like any other disciplines of basic science and engineering. Computer science is a discipline that spans theory and practice and it requires thinking both in abstract terms and in concrete terms. Nowadays, practically everyone is a computer user, and many people are even computer programmers. Computer Science can be seen on a higher level, as a science of problem solving and problem solving requires precision, creativity, and careful reasoning. The ever-evolving discipline of computer science also has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers, but finding a solution requires both computer science expertise and knowledge of the particular application domain. Computer science has a wide range of specialties. These include Computer Architecture, Software Systems, Graphics, Artificial Intelligence, Computational Science, and Software Engineering. Drawing from a common core of computer science knowledge, each specialty area focuses on specific challenges. Computer Science is practiced by mathematicians, scientists and engineers. Mathematics, the origins of Computer Science, provides reason and logic. Science provides the methodology for learning and refinement. Engineering provides the techniques for building hardware and software.

The Students completing this programme will be able to present Software application clearly and precisely, make abstract ideas precise by formulating them in the Computer languages.

Completion of this programme will also enable the learners to join teaching profession, enhance their employability for government jobs, jobs in software industry, banking, insurance and investment sectors, data analyst jobs and jobs in various other public and private enterprises.

LEARNING OUTCOMES-BASED CURRICULUM FRAMEWORK GUIDELINES BASED REGULATIONS FOR UNDER GRADUATE PROGRAMME	
Programme:	B.Sc., Information Technology
Programme Code:	
Duration:	3 years [UG]
Programme Outcomes:	PO1: Disciplinary knowledge: Capable of demonstrating comprehensive knowledge and understanding of one or more disciplines that form a part of an undergraduate programme of study PO2: Communication Skills: Ability to express thoughts and ideas effectively in writing and orally; Communicate with others using appropriate media; confidently share one's views and express herself/himself; demonstrate the ability to listen carefully, read and write analytically, and present complex information in a clear and concise manner to different groups. PO3: Critical thinking: Capability to apply analytic thought to a body of knowledge; analyses and evaluate evidence, arguments, claims, beliefs on the basis of empirical evidence; identify relevant assumptions or implications; formulate coherent arguments; critically evaluate practices, policies and theories by following scientific approach to knowledge development. PO4: Problem solving: Capacity to extrapolate from what one has learned and apply their competencies to solve different kinds of non-familiar problems, rather than replicate curriculum content knowledge; and apply one's learning to real life situations. PO5: Analytical reasoning: Ability to evaluate the reliability and relevance of evidence; identify logical flaws and holes in the arguments of others; analyze and synthesize data from a variety of sources; draw valid conclusions and support them with evidence and examples, and addressing opposing viewpoints. PO6: Research-related skills: A sense of inquiry and capability for asking relevant/appropriate questions, problem arising, synthesizing and articulating; Ability to recognize cause-and-effect relationships, define problems, formulate hypotheses, test hypotheses, analyze, interpret and draw conclusions from data, establish hypotheses, predict cause-and-effect relationships; ability to plan, execute and report the results of an experiment or investigation PO7: Cooperation/Team work: Ability to work effectively and respectfully

	with diverse teams; facilitate cooperative or coordinated effort on the part of a group, and act together as a group or a team in the interests of a common cause and work efficiently as a member of a team. PO8: Scientific reasoning: Ability to analyze, interpret and draw conclusions from quantitative/qualitative data; and critically evaluate ideas, evidence and experiences from an open-minded and reasoned perspective. PO9: Reflective thinking: Critical sensibility to lived experiences, with self-awareness and reflexivity of both self and society. PO10 Information/digital literacy: Capability to use ICT in a variety of learning situations, demonstrate ability to access, evaluate, and use a variety of relevant information sources; and use appropriate software for analysis of data. PO 11 Self-directed learning: Ability to work independently, identify appropriate resources required for a project, and manage a project through to completion. PO 12 Multicultural competence: Possess knowledge of the values and beliefs of multiple cultures and a global perspective; and capability to effectively engage in a multicultural society and interact respectfully with diverse groups. PO 13: Moral and ethical awareness/reasoning: Ability to embrace moral/ethical values in conducting one's life, formulate a position/argument about an ethical issue from multiple perspectives, and use ethical practices in all work. Capable of demonstrating the ability to identify ethical issues related to one's work, avoid unethical behavior such as fabrication, falsification or misrepresentation of data or committing plagiarism, not adhering to intellectual property rights; appreciating environmental and sustainability issues; and adopting objective, unbiased and truthful actions in all aspects of work. PO 14: Leadership readiness/qualities: Capability for mapping out the tasks of a team or an organization, and setting direction, formulating an inspiring vision, building a team who can help achieve the vision, motivating and inspiring team members to engage with that vision, and using management skills to guide people to the right destination, in a smooth and efficient way. PO 15: Lifelong learning: Ability to acquire knowledge and skills, including "learning how to learn", that are necessary for participating in learning activities throughout life, through self-paced and self-directed learning aimed at personal development, meeting economic, social and cultural objectives, and adapting to changing trades and demands of work place through knowledge/skill development/reskilling.
--	--

Highlights of the Revamped Curriculum:

- Student-centric, meeting the demands of industry & society, incorporating industrial components, hands-on training, skill enhancement modules, industrial project, project with viva-voce, exposure to entrepreneurial skills, training for competitive examinations, sustaining the quality of the core components and incorporating application oriented content wherever required.
- The Core subjects include latest developments in the education and scientific front, advanced programming packages allied with the discipline topics, practical training, devising mathematical models and algorithms for providing solutions to industry / real life situations. The curriculum also facilitates peer learning with advanced mathematical topics in the final semester, catering to the needs of stakeholders with research aptitude.
- The General Studies and Mathematics based problem solving skills are included as mandatory components in the 'Training for Competitive Examinations' course at the final semester, a first of its kind.
- The curriculum is designed so as to strengthen the Industry-Academia interface and provide more job opportunities for the students.
- The Industrial Statistics course is newly introduced in the fourth semester, to expose the students to real life problems and train the students on designing a mathematical model to provide solutions to the industrial problems.
- The Internship during the second year vacation will help the students gain valuable work experience that connects classroom knowledge to real world experience and to narrow down and focus on the career path.
- Project with viva-voce component in the fifth semester enables the student, application of conceptual knowledge to practical situations. The state of art technologies in conducting a Explain in a scientific and systematic way and arriving at a precise solution is ensured. Such innovative provisions of the industrial training, project and internships will give students an edge over the counterparts in the job market.
- State-of Art techniques from the streams of multi-disciplinary, cross disciplinary and inter disciplinary nature are incorporated as Elective courses, covering conventional topics to the latest - Artificial Intelligence.

DURATION OF THE PROGRAMME

The programme shall extend over a period of three years comprising six semesters with two semesters in one academic year. There shall not be less than 90 working days

for each semester. Examination shall be conducted at the end of every semester for the respective subjects.

COURSE OF STUDY

The programme of study shall comprise instruction in the following subjects according to the syllabus and books prescribed from time to time. The syllabus for various subjects shall be clearly demarcated into five units in each subject. Part -I, Part-II, Part- III and Part -IV subjects are prescribed in the scheme of examination. The Extension Activities are a must for each student to take part at least in any one of the activities such as NSS, YRC, SPORTS and RRC for the fulfillment of the degree.

EXAMINATIONS

The theory examination shall be three-hour duration for each paper at the end of every semester. The candidate failing in any subject(s) will be permitted to appear in the subsequent examination. The practical examinations for core subjects and SEC should be conducted at the end of every semester. Submission of record note books for practical examinations candidates appearing for practical examinations should submit bonafide Record note books prescribed for practical examinations, Otherwise the candidates will not be permitted to appear for the practical examinations. However, in genuine cases of the students who could not submit the record note books, they may be permitted to appear for the practical examinations, provided the concerned Head of the Department certify that the candidate has performed the experiments prescribed for the course. For such candidates zero (0) marks will be awarded for record note books.

REVISION OF REGULATIONS AND CURRICULUM

The University may revise/amend/ change the Regulations and Scheme of Examinations, as and when found necessary.

COMMENCEMENT OF THIS REGULATION:

These regulations shall take effect from the academic year 2026-2027, i.e., for students who are to be admitted to the first year of the programme during the academic year 2026-2027 and thereafter.

PATTERN OF QUESTION PAPER

Time: Three Hours

Maximum: 75 marks

Part A- (15X1 = 15marks)

Answer all questions

(a),(b),(c),(d) Multiple Choice

Part B-(2X5 = 10 marks)

Answer any Two questions out of five

(One question from each unit)

Part C- (5X10 = 50 marks)

Answer ALL questions

Either (or) Type Five questions

(One question from each unit)

INTERNAL MARKS AND EXTERNAL MARKS

	Internal Marks	External Marks
Theory	25	75
Practical	40	60
Project	25	75
Internship/Health & Wellness	Commended/Highly Commended	

(a) Practical-External marks distribution (Total Marks: 60)

For each practical question the marks should be awarded as follows (External)

- i) Algorithm/flowchart - 20%
- ii) Writing the program in the main answer book - 30%
- iii) Test and debug the program - 30%
- iv) Printing the correct output - 20%

(Marks may be proportionately reduced for the errors committed in each of the above)

Practical Question Paper Pattern

Student should attend two questions (either or pattern)

COURSE OF STUDY AND SCHEME OF EXAMINATION

SNO	Sem	Paner code	Part	Course Group	Paper Type	Subject Title	Hours Week	Exam Hours	Internal Marks	External Marks	Total Marks	Credit
1	I		I	Foundation	Theory	Language–Tamil	6	3	25	75	100	3
2			II	Foundation	Theory	Language– English	6	3	25	75	100	3
3			III	Core	Theory	Problem Solving Using C++	6	3	25	75	100	5
4			III	Core	Practical	Practical: Programming in C++ Lab	3	3	40	60	100	5
5			III	Allied	Theory	Maths/Physics/Electronics/Statistics	6	3	25	75	100	4
6			IV	NME-I	Theory		2	3	25	75	100	2
7			IV	Value Added	Theory	Value Education	1	3	25	75	100	1
Total					7		30					23
8	II		I	Foundation	Theory	Language–Tamil	6	3	25	75	100	3
9			II	Foundation	Theory	Language– English	6	3	25	75	100	3
10			III	Core	Theory	Data Structures and Algorithms	5	3	25	75	100	5
11			III	Core	Practical	Practical: Data Structures using C++ Lab	3	3	40	60	100	5
12			III	Allied	Theory	Maths/Physics/Electronics/Statistics	3	3	25	75	100	2
13			III	Allied	Practical	Maths/Physics/Electronics/Statistics	2	3	40	60	100	2
14			IV	NME-II	Theory		2	3	25	75	100	2
15			IV	Value Added	Theory	Disaster Management	1	3	25	75	100	1
16					Naan Mudhalvan	NMSD	2		25	75	100	2
Total					7+1		30					23+ 2

Allied Subjects (Colleges can choose any one of the paper as Allied per year)

1. Mathematics 2. Physics 3. Statistics 4. Electronics

* Naan Mudhalvan – Courses – external 25 marks will be assessed by industry and internal will be offered by respective course teacher. Naan Mudhalvan reappear students – Alternate SKILL papers arranged

Problem Solving using C++	
Learning Objectives	
LO1	Understand the fundamentals of problem solving, algorithms, flowcharts, and object-oriented programming concepts.
LO2	Apply C++ programming constructs such as data types, operators, expressions, and control structures in developing programs.
LO3	Design and implement classes, objects, constructors, destructors, and member functions using object-oriented principles
LO4	Demonstrate inheritance, operator overloading, polymorphism, and virtual functions in software development.
LO5	Develop C++ applications using pointers, file handling, and advanced object-oriented programming techniques.
CONTENT	
UNIT I: Introduction: Steps Involved in Problem Solving Using Computers- Characteristics of Algorithms- Symbols Used in Flow Charts - Pseudocode: Sequence, Selection and Iteration - Principles of Object-Oriented Programming: Basic Concepts of Object-Oriented Programming-Benefits of OOP- Applications of OOP - Beginning with C++: Simple C++ Program-Structure of C++ Program- Tokens, Expressions and Control Structures: Introduction - Tokens- Keywords - Identifiers and Constants - Basic Data Types - User Defined Data Types - Derived Data Types -Symbolic Constants	
UNIT II : Declaration of Variables - Operators in C++ - Scope Resolution Operator - Type Cast Operator - Expression and their Types - Assignment Expressions - Implicit Conversion - Operator Overloading - Operator Precedence - Control Structures- Function in C++ : Main Function - Function Prototyping - Call by Reference - Return by Reference - Inline Functions - Function Overloading - Friend and Virtual Functions - Math Library Functions.	
UNIT III: Classes and Objects: Specifying Class - Defining Member Functions - Nesting of Member Functions - Arrays within a Class - Memory Allocation for Objects - Static Data Members- Static Member Functions - Arrays of Objects - Object as Function Arguments - Friendly Functions - Returning Objects - const Member Functions - Constructors and Destructors: Constructors - Multiple Constructors in a Class -Constructors with Default Arguments - Dynamic Initialization of Objects - Copy Constructor - Dynamic Constructors - Destructors.	
UNIT IV: Operator Overloading and Type Conversion: Defining Operator Overloading - Overloading Unary Operators - Overloading Binary Operators - Rules for Overloading Operators- Type Conversions - Inheritance: Extending Classes: Defining Derived Classes - Single Inheritance - Multilevel Inheritance - Multiple Inheritance - Hierarchical Inheritance - Hybrid Inheritance - Virtual Base Classes - Abstract Classes.	
UNIT V: Pointers, Virtual Functions and Polymorphism: Introduction - Pointers- Pointers to Objects - Pointers to Derived Classes - Virtual Functions - Pure Virtual Functions -	

Working with Files: Classes for File Stream Operations - Opening and Closing a File - Detecting end-of-file - More about Open(): File Modes- Sequential Input and Output Operations - Updating a file: Random Access - Command-line Arguments.	
Course Outcomes	
CO1	Explain problem-solving techniques, algorithms, flowcharts, and the principles of object-oriented programming.
CO2	Develop C++ programs using operators, expressions, functions, and control structures.
CO3	Design programs using classes, objects, constructors, destructors, and member functions.
CO4	Apply inheritance, polymorphism, operator overloading, and type conversion concepts in application development.
CO5	Implement file handling, pointers, virtual functions, and dynamic binding concepts in C++ programs.
Learning Resources	
Text Books	1. "Problem Solving With C" Somashekara, M. T.,Guru, D. S., Manjunatha, K.S. PHI Learning Pvt. Ltd.,[UNIT -I -First Half] 2. Object Oriented Programming With C++,E. BalagurusamyTataMcGraw-Hill, NewDelhi, 4thEdition
Reference Books	1. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education 2. Programming: Principles and Practice Using C++, 3rd Edition, Bjarne Stroustrup Addison-Wesley Professional Publisher. 3. C++: The Complete Reference- Schildt, McGraw-Hill Education (India)
Website/ Link	https://www.w3schools.com/cpp/cpp_oop.asp https://www.geeksforgeeks.org/cpp/object-oriented-programming-in-cpp/

Mapping with Programme Outcomes

CO/PSO	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO 1	3	2	1	1	1
CO 2	3	3	2	1	2
CO 3	2	3	3	2	2
CO 4	2	3	3	3	2
CO 5	2	2	3	3	3
Weightage of course contributed to each PSO	12	13	12	10	10

S-Strong-3 M-Medium-2 L-Low-1

Programming in C++ Lab	
Learning Objectives	
LO1	Understand the fundamentals of problem solving, algorithms, flowcharts, and object-oriented programming concepts.
LO2	Apply C++ programming constructs such as data types, operators, expressions, and control structures in developing programs.
LO3	Design and implement classes, objects, constructors, destructors, and member functions
LO4	Demonstrate inheritance, operator overloading, polymorphism, and virtual functions in software development.
LO5	Develop C++ applications using pointers, file handling, and advanced object-oriented programming techniques.
LAB EXERCISES	
1. Write a C++ program to find the sum of the digits of a given number. 2. Write a C++ program to find the largest among three numbers using if-else. 3. Write a C++ program to make a simple calculator that can perform arithmetic operations. 4. Write a C++ program to find the factorial of a given number using recursion function. 5. Write a C++ program to demonstrate the usage of a constructor and destructor to manage the bank account records. 6. Write a C++ program to demonstrate polymorphism by calculating area of rectangle and triangle using a shape class. 7. Write a C++ program using Binary operator (+, -, *, /) for Operator Overloading . 8. Write a C++ program using inheritance where Base class as “Student details” and Derived class as “Marks and Grade calculation” 9. Write a C++ program to reverse a given string using pointers. 10. Write a C++ program to count the number of characters in a file.	
Course Outcomes	
CO 1	Explain problem-solving techniques, algorithms, flowcharts, and the principles of object-oriented programming.
CO 2	Develop C++ programs using operators, expressions, functions, and control structures.
CO 3	Design programs using classes, objects, constructors, destructors, and member functions.
CO 4	Apply inheritance, polymorphism, operator overloading, and type conversion concepts in application development.

CO 5	Implement file handling, pointers, virtual functions, and dynamic binding concepts in C++ programs.
-------------	---

Mapping with Programme Outcomes

CO/PSO	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO 1	3	2	1	1	1
CO 2	3	3	2	1	2
CO 3	2	3	3	2	2
CO 4	2	3	3	3	2
CO 5	2	2	3	3	3
Weightage of course contributed to each PSO	12	13	12	10	10

S-Strong-3 M-Medium-2 L-Low-1

Data Structures and Algorithms	
Learning Objectives	
LO1	Understand abstract data types and complexity analysis.
LO2	Develop problem-solving and logical thinking skills.
LO3	Design and implement efficient algorithms.
LO4	Apply DSA concepts using Python/C.
LO5	Apply DSA in real-world problem solving.
CONTENT	
UNIT I: Introduction and Overview: Definitions - Overview of Data Structures - Implementation of Data Structures - Abstract Data Types. Arrays: One Dimensional Array - Multidimensional Arrays.	
UNIT II: Linked Lists: Single Linked List - Double Linked List - Circular Linked List - Circular Double Linked List. Stacks: Stacks: Representation of Stack - Array Representation of Stacks – Operations on Stacks. Applications of Stacks: Evaluation of Arithmetic Expressions - Implementation of Recursion - Factorial Calculation.	
UNIT III: Queues & Applications: Definition - Representation of Queues. Various Queue Structures: Circular Queue – Deque - Priority Queue. Application of Queues: Round Robin Algorithm. Trees: Trees: Basic Terminologies - Binary Trees. Operations on Binary Tree: Insertion – Deletion – Traversals. Types of Binary Trees: Expression Tree - Binary Search Tree - Heap Trees.	
UNIT IV: Graphs: Definitions - Graph Representations - Elementary Graph Operations: Depth First Search - Breadth First Search - Spanning Trees. Minimum Cost Spanning Trees: Kruskals Algorithm - Prims Algorithm. Sorting : Basic Terminologies - Selection Sort - Insertion Sort - Bubble Sort - Shell sort - Quick Sort and Merge Sort.	
UNIT V: Searching: Linear Search Techniques -Linear Search with Array - Linear Search with Linked List. Non -Linear Search Techniques: Binary Tree Searching - Binary Search Tree Searching. Hashing: Introduction.	
CourseOutcomes	
CO1	Understand abstract data types, Apply linear data structures such as arrays.
CO2	Apply linked lists to solve computational problems.
CO3	Apply linear data structures such as queues to solve computational problems. Design and implement tree.
CO4	Design and implement graph structures, and perform traversal techniques like BFS and DFS, Analyze the sorting
CO5	Apply searching, and hashing techniques to efficiently organize and retrieve data.

	LearningResources
Text Books	1. Debasis Samanta “Classic Data Structures” PHI Learning Pvt Ltd, Second Edition, 2018. 2. Ellis Horowitz, Sarataj Sahni, Susan Anderson-Freed “Fundamentals of Data Structures in C” University Press, Second Edition, 2023.
Reference Books	1. A.V. Aho, J.D. Ullman, J.E. Hopcraft ” Data Structures and Algorithms”, Pearson Education. 2. Seymour Lipschutz, Data Structures McGraw Hill Publications, 2014, 1st Edition. 3. A Chitra, P T Rajan “Data Structures”, Vijay Nicole Imprints Private Limited, 2016. 4. Reema Thareja, “Data Structures Using C” Oxford University Press, Third Edition 2023. 5. Samanta.D , Classic Data Structure Prentice Hall of India Pvt Ltd 2007, 9th Edition.
Website/ Link	1. https://www.geeksforgeeks.org/datastructures/ 2. https://www.tutorialspoint.com/data_structures_algorithms/index.htm 3. https://www.worldscientific.com/worldscibooks/10.1142/5256#t=aboutBook

DATA STRUCTURES USING C ++ Lab

CO/PSO	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO 1	3	3	2	3	3
CO 2	3	2	3	2	2
CO 3	3	3	2	2	2
CO 4	3	2	3	3	2
CO 5	3	3	3	2	3
Weightage of course contributed to each PSO	15	13	13	12	12

S-Strong-3 M-Medium-2 L-Low-1

Learning Objectives	
LO1	To understand the concepts of ADTs
LO2	To learn linear data structures-lists, stacks, queues
LO3	To learn Tree structures and application of trees
LO4	To learn graph structures and application of graphs
LO5	To understand various sorting and searching
LAB EXERCISES	
1. Write a C++ program to implement the LIST ADT using arrays. 2. Write a C++ program to multiply two matrices A and B and store the resultant matrix in C using arrays. 3. Write a C++ program to experiment the operation of STACK using array. 4. Write a C++ program to create menu driven options to implement QUEUE to perform the following (i) Insertion (ii) Deletion (iii) Modification (iv) Listing of elements 5. Write a C++ program to create Linked list representations of employee records and do the following operations using pointers. (i) To add a new record. (ii) To delete an existing record. (iii) To print the details about an employee. (iv) To find the number of employees in the structure. 6. Write a C++ program that uses stack Operations to perform converting infix expression into postfix expression. 7. Write a C++ program to create a binary tree using Function to perform the Traverse of the binary tree in Preorder, inorder and postorder. 8. To write a C++ program to implement graph traversals by Breadth First Search and Depth First Search. 9. Write a C++ program for implementing the following searching methods i) Linear Search ii) Binary Search 10. Write a C++ program that implements the following sorting methods to sort a given list of integers in ascending order i) Bubble sort ii) Selection sort	
Course Outcomes	
CO 1	Understand basic data structures such as arrays

CO 2	Analysis the use of pointers for linked list, doubly linked list
CO 3	Understand basic data structures such stacks and queues
CO 4	Solve problem involving trees and graphs
CO 5	Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data.

Mapping with Programme Outcomes

CO/PSO	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO 1	3	2	1	1	1
CO 2	3	3	2	1	2
CO 3	2	3	3	2	2
CO 4	2	3	3	3	2
CO 5	2	2	3	3	3
Weightage of course contributed to each PSO	12	13	12	10	10

S-Strong-3 M-Medium-2 L-Low-1