POORNAPRAJNA COLLEGE (AUTONOMOUS), UDUPI

NAAC Re-Accredited "A+" (3.27 CGPA)

(Promoted and managed by Udupi Shree Adamaru Matha Education Council, Bengaluru)

SYLLABUS FOR UNDERGRADUATE PROGRAM (UG) OF COMPUTER SCIENCE CURRICULUM FRAMEWORK

COURSE PATTERN AND SCHEME OF EXAMINATION FOR UG COURSES

FRAMED ACCORDING TO THE STATE EDUCATION POLICY (SEP 2024)

BSC COMPUTER SCIENCE
I & II SEMESTERS

TO IMPLEMENT FROM THE ACADEMIC YEAR 2025-26

BOARD OF STUDIES IN COMPUTER SCIENCE
POORNAPRAJNA COLLEGE (AUTONOMOUS),
UDUPI - 576101

Preamble

Computer Science (CS) has been evolving as an important branch of science and technology in last two decade and it has carved out a space for itself like computer science and engineering. Computer science spans theory and more application and it requires thinking both in abstract terms and in concrete terms.

The ever -evolving discipline of computer science has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers and its applications, 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, Mathematical and Statistical Analysis, Data Science, Computational Science, and Software Engineering.

In India, it was initially introduced at the Master (postgraduate) level as MCA and M.Tech. Later on, engineering programmes such as B.Tech and B.E in Computer Science & Engineering and in Information Technology were introduced in various engineering College/Institutions to cater to the growing demand for trained engineering manpower in IT industries. Parallelly, BCA, BSc and MSc programmes with specialization in Computer Science were introduced to train manpower in this highly demanding area.

BSc.aimed at undergraduate level training facilitating multiple career paths. Students so graduated, can take up postgraduate programmes in CS or MCA leading to research as well as R&D, can be employable at IT industries, or can pursue a teaching profession or can adopt a business management career.

B.Sc. aims at laying a strong foundation of computer application at an early stage of the career. There are several employment opportunities and after successful completion of B.Sc. graduating students can fetch employment directly in companies as programmer, Web Developer, Software Engineer, Network Administrator, Data Scientist, or AI/ML personnel.

The Program outcomes in B.Sc. are aimed at allowing flexibility and innovation in design and development of course content, in method of imparting training, in teaching learning process and in assessment procedures of the learning outcomes. The emphasis in B.Sc. courses, in outcome-based curriculum framework, help students learn solving problems, accomplishing IT tasks, and expressing creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages.

All students must, therefore, have access to a computer with a modern programming language installed. The computer science framework does not prescribe a specific language. The teacher and students will decide which modern programming languages students will learn. More importantly, students will learn to adapt to changes in programming languages and learn new languages as they are developed. The present Curriculum Framework for B.Sc. degree is intended to facilitate the students to achieve the following.

- To develop an understanding and knowledge of the basic theory of Computer Science and Information Technology with good foundation on theory, systems and applications such as algorithms, data structures, data handling, data communication and computation

- To develop the ability to use this knowledge to analyse new situations in the application domain

- To acquire necessary and state-of-the-art skills to take up industry challenges. The objectives and outcomes are carefully designed to suit to the above-mentioned purpose.

- The ability to synthesize the acquired knowledge, understanding and experience for a better and improved comprehension of the real-life problems

- To learn skills and tools like mathematics, statistics and electronics to find the solution, interpret the results and make predictions for the future developments

- To formulate, to model, to design solutions, procedure and to use software tools to solve real world problems and evaluate

The objectives of the Programme are:

- The primary objective of this program is to provide a foundation of computing principles and business practices for effectively using/managing information systems and enterprise software.
- It helps students analyze the requirements for system development and exposes students to business software and information systems
- This course provides students with options to specialize in legacy application software, system software or mobile applications
- To produce outstanding IT professionals who can apply the theoretical knowledge into practice in the real world and develop standalone live projects themselves
- To provide opportunity for the study of modern methods of information processing and its applications.
- To develop among students the programming techniques and the problem- solving skills through programming.
- To prepare students who wish to go on to further studies in computer science and related subjects.
- To acquaint students to Work effectively with a range of current, standard, Office Productivity software applications

Program Outcomes: B.Sc. (3 Years) Degree

- Discipline knowledge: Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity
- Problem Solving: Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.
- Design and Development of Solutions: Ability to design and development of algorithmic solutions to real world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems.
- Programming a computer: Exhibiting strong skills required to program a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.
- Application Systems Knowledge: Possessing a sound knowledge on computer application software and ability to design and develop app for applicative problems.
- Modern Tool Usage: Identify, select and use a modern scientific and IT tool or technique for modeling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.
- Communication: Must have a reasonably good communication knowledge both in oral and writing. Project Management: Practicing of existing projects and becoming independent to launch own project by identifying a gap in solutions.
- Ethics on Profession, Environment and Society: Exhibiting professional ethics to maintain the integrality in a working environment and also have concern on societal impacts due to computer-based solutions for problems.
- Lifelong Learning: Should become an independent learner. So, learn to learn ability.
- Motivation to take up Higher Studies: Inspiration to continue education towards advanced studies on Computer Science**.**

CURRICULUM STRUCTURE FOR I TO II SEMESTER BSC-COMPUTER SCIENCE

Semester I								
Sl. No.	Course Code	Title of the Course	Category of Courses	Teaching Hours per Week	SEE	IA	Total Marks	Credits
1		Language – I	Lang	4	80	20	100	3
2		Language – II	Lang	4	80	20	100	3
3	BSC-CAT-1.1	SCIENCE-1	Core	4	80	20	100	3
4	BSC-CAT-1.2	SCIENCE--2	Core	4	80	20	100	3
5	BSCSCS101	SCIENCE-3	Core	4	80	20	100	3
6	BSC-CAP-1.1	SCIENCE-1	Practical	4	40	10	50	2
7	BSC-CAP-1.1	SCIENCE-2	Practical	4	40	10	50	2
8	BSCSPS101	SCIENCE-3	Practical	4	40	10	50	2
9		Constitution/ Values	Compulsory	2	40	10	50	2
Sub –Total				34	560	140	700	23

Semester II								
Sl. No.	**Course Code**	**Title of the Course**	**Category of Courses**	**Teaching Hours per Week**	**SEE**	**IA**	**Total Marks**	**Credits**
1		**Language – I**	**Lang**	**4**	**80**	**20**	**100**	**3**
2		**Language – II**	**Lang**	**4**	**80**	**20**	**100**	**3**
3	**BSC-CAT-1.1**	**SCIENCE-1**	**Core**	**4**	**80**	**20**	**100**	**3**
4	**BSC-CAT-1.2**	**SCIENCE-2**	**Core**	**4**	**80**	**20**	**100**	**3**
5	**BSCSCS201**	**SCIENCE-3**	**Core**	**4**	**80**	**20**	**100**	**3**
6	**BSC-CAP-1.1**	**SCIENCE-1**	**Practical**	**4**	**40**	**10**	**50**	**2**
7	**BSC-CAP-1.1**	**SCIENCE--2**	**Practical**	**4**	**40**	**10**	**50**	**2**
8	**BSCSPS201**	**SCIENCE-3**	**Practical**	**4**	**40**	**10**	**50**	**2**
9		**Constitution/ Values**	**Compulsory**	**2**	**40**	**10**	**50**	**2**
Sub –Total				**34**	**560**	**140**	**700**	**23**

CURRICULUM STRUCTURE FOR I SEMESTER BSc- COMPUTER SCIENCE

Semester I								
Sl. No	**Course Code**	**Title of the Course**	**Category of Courses**	**Teaching Hours per Week**	**SEE**	**IA**	**Total Marks**	**Credits**
1	**BSCSCS101**	Computer Fundamentals and Programming in C	**Core**	**4**	**80**	**20**	**100**	**3**
2	**BSCSPS101**	C Programming Lab	**Practical**	**4**	**40**	**10**	**50**	**2**

Program Name	**BSC-COMPUTER SCIENCE**	Semester	**I**
Course Title	**Computer Fundamentals and Programming in C (Theory)**		
Course Code:	**BSCSCS101**	No. of Credits	**03**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Course Outcomes (COs)

After completing this course satisfactorily, a student will be able to:

- **Operate desktop computers to carry out computational tasks**
- **Understand working of hardware and software and the importance of operating systems**
- **Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts**
- **Read, understand and trace the execution of programs written in C language**
- **Write the C code for a given problem**
- **Perform input and output operations using programs in C**
- **Write programs that perform operations on arrays**

Topics	**Book**	**Chapter/Page No./ Section No.**
UNIT I		
Fundamentals of Computers: Introduction to Computers –Definition of a computer, Characteristics of computers Generations of computers, Classification of computers. Computer system, applications of computers. **Computer Software** – Categories of software. Computer Programming and Languages–Machine Level, Assembly level and High-level languages; Translator Programs – Assembler, Interpreter and Compiler. Developing a computer program,	Book 1 Book 1	Chapter 1: All sections (sub sections included) Chapter 11: sections 11.1, 11.2,

Program Development Cycle - Algorithm, Flowchart and Pseudocode with examples. **Introduction to C Programming**: Overview of C; History and Importance of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C. **C Programming Basic Concepts**: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration and initialization of variables; Symbolic constants.	Book 2 Book 2	11.3 Chapter 10: 10.1, 10.2, 10.3, 10.5 Chapter 1 Chapter 2
UNIT II		
C Operators and Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associatively. Evaluation of arithmetic expressions; Type conversion. **Input and output with C**: Formatted I/O functions - printf and scanf, control stings and escape sequences, output specifications with printf functions; Unformatted I/O functions to read and display single character and a string - getchar, putchar, gets and puts functions. **Control Structures**: Branching: if, if-else, nested if, else-if ladder, switch. **Looping:** while, do-while and for loop, nested loops, exit, break, jumps in loops.	Book 2	Chapter 3 Chapter 4 Chapter 5 Chapter 6
UNIT III		

Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation. **Handling of character Strings**: Declaring and Initializing string variables; String handling functions - strlen, strcmp, strcpy and strcat; Character handling functions - toascii, toupper, tolower, isalpha, isnumeric etc. **User-Defined Functions:** Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type. Nesting of functions, Recursion, and functions with arrays, the scope, visibility & lifetime of variables (Storage classes).	Book 2	Chapter 7 Chapter 8 Chapter 9
UNIT IV		
Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions. **Pointers in C:** Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic. **File Management in C:** Basic file operations, types of files. Creating text file. Modes of opening a file, formatted and unformatted i/o operations. **The Preprocessor:** Macro substitution and file inclusion.	Book 2	Chapter 10 Chapter 11 Chapter 12 Chapter 14

Textbooks:

1. ITL Education Solution Limited, Introduction to Information Technology, Second Edition 2018, Pearson Education.
2. E. Balagurusamy: Programming in ANSI C (TMH),7th Edition.

Reference Books :

1. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
2. V. Rajaraman: Programming in C (PHI – EEE)
3. S. Byron Gottfried: Programming with C (TMH)
4. Kernighan &Ritche: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)
7. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication.

C Programming Lab

Program Name	**BSC-COMPUTER SCIENCE**	Semester	**I**
Course Title	**C Programming Lab**		
Course Code:	**BSCSPS101**	No. of Credits	**02**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART – A

1. Program to check for prime number.
2. Program to generate n Fibonacci numbers.
3. Program to read a multi – digit number find the sum of the digits, reverse the number and check it for palindrome.
4. Program to find the roots of quadratic equation (Demonstration of switch Statement).
5. Program to count occurrences of each character in a string.
6. Program to read ‘n’ integer values into a single dimension array and arrange them in ascending order using bubble sort method

7. Program to find the largest and smallest elements with their position in a one-dimensional array.

8. Program to multiply two matrices.

PART – B

1. Program to accept 'n' and find the sum of the series 1! + 3! + 5! …… + n!

2. Write user-defined functions to (a) find the length of a string (b) concatenate two strings. Call these functions in the main program

3. Program to find whether a given string is palindrome or not (Use a function to reverse a string using pointers)

4. Program to add two matrices using pointers.

5. Program to create a file NUMBER.txt, separates Odd and Even numbers and copy to file ODD.txt and EVEN.txt respectfully. Finally display all the three files.

6. Program to find the GCD of 'n' integers using a function to compute the GCD of two integers

7. Program to enter the information of n students (name, register number, marks in three subjects) into an array of structures. Compute and print the result of all students. For passing, student should get at least 35 in each subject, otherwise result is "FAIL". If the student passes and if percentage >= 70, result is DISTINCTION; If percentage is < 70 and >=60, result is FIRST CLASS; if percentage is < 60 and >=50, result is SECOND CLASS; otherwise, result is PASS CLASS. Get the output of all students in a tabular form with proper column headings

8. Program to prepare the pay slip of 'n' employees using an array of structures. Input the employee's name, employee number and basic pay. Calculate the DA,HRA, PF, PT, Gross Pay and Net Pay as follows: If Basic < 40000, DA = 50% of Basic, HRA = 12% of Basic, PF = 12% of Gross Pay, PT =250. Otherwise, DA = 40% of Basic, HRA = 10% of Basic, PF = 13% of Gross, PT= 300. Gross Pay = Basic + DA + HRA and Net Pay = Gross Pay – PF – PT

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	**PART-A** **Writing:7 Marks Execution:8Marks**	**15Marks**
Program-2	**PART-B** **Writing:10 Marks** **Execution:10Marks**	**20 Marks**
Practical Record		**05 Marks**

Total	**40 Marks**

CURRICULUM STRUCTURE FOR II SEMESTER BSc- COMPUTER SCIENCE

Semester II								
Sl. No	**Course Code**	**Title of the Course**	**Category of Courses**	**Teaching Hours per Week**	**SEE**	**IA**	**Total Marks**	**Credits**
1	**BSCSCS201**	Data Structures	**Core**	**4**	**80**	**20**	**100**	**3**
2	**BSCSPS201**	Data Structures Lab	**Practical**	**4**	**40**	**10**	**50**	**2**

Program Name	**BSC-COMPUTER SCIENCE**	Semester	**II**
Course Title	**Data Structures (Theory)**		
Course Code:	**BSCSCS201**	No. of Credits	**03**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Course Outcomes (COs)

After completing this course satisfactorily, a student will be able to:

- **Learn the basic types for data structure, implementation and application.**
- **Know the strength and weakness of different data structures.**
- **Use the appropriate data structure in context of solution of given problem.**
- **Develop programming skills which require to solve given problem.**

Topics	Book	Chapter/Page No./ Section No.
UNIT 1 [13 Hours]		
Introduction to Data Structures: Definition, Need for Data Structures, Types of Data Structures. **Linear Data Structures: Arrays** - Definition, Declaration and storage of one-and two-dimensional arrays. Sparse matrices. **Recursion:** Definition; Types of recursions; Recursion Technique Examples – Fibonacci numbers, GCD, Binomial coefficient n_{Cr}, Comparison between iterative and recursive functions. **Sorting:** Sorting Selection sort, Bubble sort, Quicksort, Insertion sort; Comparison of Different sorting techniques.	BOOK-1	CHAPTER - 1 1.1 to 1.4 CHAPTER - 4 4.1,4.2,4.4,4.5,4.6,4.10,4.17 CHAPTER – 6 6.8(complexity excluded) 4.7,6.7,9.1,9.3,9.4(complexity excluded)
UNIT 2 [13 Hours]		
Searching: Introduction, Linear search, Binary Search, Comparison of different searching techniques. **Dynamic memory allocation**: Static and Dynamic memory allocation; Memory allocation and deallocation functions- *malloc,calloc,realloc* and *free*. **Linked List:** Introduction, characteristics, types of linked lists, Representation of singly linked list in memory, Singly linked list – Operations, algorithms, Representation of polynomials using linked lists. Circular linked list–Operations, Doubly linked list-operations .Memory allocations.	BOOK-1	CHAPTER – 4 4.8(complexity excluded in both 4.8 & 4.9),4.9. CHAPTER - 4 4.13 CHAPTER - 5 5.1,to 5.11,
UNIT 3 [13 Hours]		
Stacks: Array representation of stacks, Linked representation of stacks, operations, Applications of stacks Recursion, Implementation of recursive procedure by stack (Factorial function and Fibonacci sequence). **Arithmetic Expressions:** Prefix, infix and	BOOK-1	CHAPTER - 6 6.1,6.2,6.3,6.4,6.6,6.7,6.10 6.11,6.12,6.14,6.15,6.16(page number 6.79,6.80,6.87

postfix notation, infix to postfix conversion, evaluation of postfix expression. **Queues:** Array representation of queue, Linked representation of queue, Types of queues-Simple queue, circular queue, double-ended queue, priority queue, operations on queues.		Sonly),
UNIT 4 [13 Hours]		
Trees: Definition; Tree terminologies – node, root node, parent node, ancestors of anode, siblings, terminal and non-terminal nodes, degree of a node, level, edge, path, depth; **Binary tree:** Types of binary trees-strict binary tree, complete binary tree, binary search tree. Array representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal; Construction of a binary tree when inorder and pre/postorder traversals are given. **Graphs:** Terminologies, Matrix representation of graphs, Traversals: Breadth First Search and Depth first search.	Book 3	CHAPTER - 7 7.1,7.2,7.3,7.4,7.5,7.8,7.9 CHAPTER - 8 8.1,8.2,8.3,8.5,8.7

Textbooks:

1. Seymour Lipschutz, Schaum's Outlines Series, Data Structures with C, TataMcGrawHill2011

Reference Books:

1. Sartaj Sahni: Fundamentals of Data Structures.
2. Yedidyah Langsam, MosheJ.Augenstein and Aaron M. Tenenbaum, Data Structures Using C and C++,2ndEdition, PHI Publication
3. Kamathane: Introduction to Data structures (Pearson Education)
4. Y.Kanitkar: Data Structures Using C(BPB)
5. Kottur: Data Structure Using C
6. Padma Reddy: Data Structure Using C
7. Sudipa Mukherjee: Data Structures using C 1000 Problems and Solutions (McGrawHillEducation,2007)

	SCIENCE		
Course Title	**Data Structures Lab**		
Course Code:	**BSCSPS201**	No. of Credits	**02**
Contact hours	**4 Hours per week**	Duration of SEA/Exam	**3hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART-A

1. Program to compute power of a number using a recursive function.

2. Program to read the names of the cities and arrange them alphabetically.

3. Program to sort the given list using selection sort technique.

4. Program to find the sum of n elements entered by user. Use the dynamic memory to store the data and malloc () function.

5. Program to implement stack using arrays.

6. Program to sort the given list using insertion sort technique.

7. Program to search an element using linear search technique.

8. Program to search an element using recursive binary search technique.

PART-B

1. Program to implement queue using arrays.

2. Program to sort the given list using quick sort technique.

3. Write a Program for converting an Infix Expression to Postfix Expression. Program should support both parenthesized and free parenthesized expressions with the following operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.

4. Program to implement all operations on a sorted singly linked list.

5. Program to implement queue using linked list.

6. Program to implement circular queue using array.

7. Write a menu driven program for the following operations on Binary Search Tree (BST) of integers (a) Create a BST of ‘n’ Integers. (b) Traverse the BST in Inorder, Preorder and Postorder. (c) Search the BST for a given element and report the appropriate message.

8. Program for the following operations on a graph (G) of cities: (a) Create a graph of N cities using Adjacency Matrix. (b) Print all the nodes reachable from a given starting node in a digraph using BFS method.

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	**PART-A** **Writing:7 Marks Execution:8Marks**	**15Marks**
Program-2	**PART-B** **Writing:10 Marks** **Execution:10Marks**	**20 Marks**
Practical Record		**05 Marks**
Total		**40 Marks**

Questions Paper Pattern for Core Subjects

Duration:3 Hours	**Question Pattern**	Max.Marks:80
Note: Answer any ten Questions from Part-A. And one full Questions from each unit in Part-B		
Part-A		
Sub Question	**Unit**	**Marks**
a,b,c,	**1**	10*2=20
d,e,f,	**2**	
g,h,i	**3**	
j,k,l	**4**	
Part-B **Each main questions contain 3 or more sub question.**		
Unit- 1		**15**
2.		
3.		
Unit- 2		**15**
4.		
5.		
Unit- 3		**15**
6.		
7.		
Unit- 4		**15**
8.		
9.		
Total		**80**