

B.E. in Computer Science and Engineering Choice Based Credit System (CBCS) applicable for 2024 Scheme SEMESTER – I / II				
Principles of Programming Using C (2:0:2) 3 (Effective from the academic year 2024-25)				
Course Code	BPOP13/23	CIE Marks	50	
Teaching Hours/Week (L:T:P)	2:0:2	SEE Marks	50	
Total Number of Contact Hours	26 hours of theory + 14 hours of Practical	Exam Hours	3	
Course Objectives: This course will enable students to: <ul style="list-style-type: none">● Gain proficiency in the syntax and semantics of the C programming language.● Understand the structure and components of a C program, including functions, data types, operators, and control & Looping structures.● Learn to analyse problems, design algorithms, and implement solutions using C.● Apply C programming skills to real-world problems and projects.				
Preamble: In the ever-evolving landscape of technology and computer science, understanding the foundational principles of programming is crucial for aspiring developers and engineers. The C programming language, known for its efficiency, versatility, and influence on many modern languages, serves as an excellent medium for learning these principles. This course, "Principles of Programming Using C," aims to equip students with a solid understanding of core programming concepts, including algorithms and data structures. Through a combination of theoretical knowledge and practical exercises, students will learn to write efficient, maintainable, and robust code. The course will emphasize problem-solving techniques and critical thinking, fostering a mindset of continuous learning and adaptation. By the end of the course, students will have a strong foundation in programming that can be applied to real time problems and advanced study in computer science.				
Module 1 Basic Definitions of programming: Algorithm, Flowchart, Programs. Introduction To The C Language: Structure of a C Program, Your First C Program, Comments, The Greeting Program, Identifiers, Types, Void Type, Integral Type, Floating-Point Types. Variables: Variable Declaration, Variable Initialization Constants: Constant Representation, Coding Constants Input/Output: Streams, Formatting Input/Output Textbook 1: Chapter 1(1.1 to 1.4), Chapter 2(2.1-2.7) (05 Hours)				
Module 2 Type Conversion: Implicit Type Conversion, Explicit Type Conversion Logical Data and Operators: Logical Data in C, Logical Operators, Evaluating Logical Expressions, Comparative Operators Two-Way Selection: if... else, Null else Statement, Nested if Statements, Dangling else Problem, Simplifying if Statements, Conditional Expressions. Multway Selection: The switch Statement, The else-if Concept of a loop: Pretest and Post-test Loops, Initialization and Updating, Loop Initialization, Loop Update Loops in C: The while Loop, The for Loop, The do....while Loop, The Comma Expression, Other Statements Related to Looping Textbook 1: Chapter 3 (3.5 to 3.6), Chapter 5 (5.1 to 5.3), Chapter 6 (6.1 to 6.7) (05 Hours)				

<p style="text-align: center;">Module 3</p> <p>FUNCTIONS: Designing Structured Programs, Functions in C User-Defined Functions: Basic Function Designs, Function Definition, Function Declaration, The Function Call, Function Examples Inter-Function Communication: Basic Concept, C Implementation. Scope: Global Scope, Local Scope. Recursion: Iterative Definition, Recursive Definition, Iterative Solution, Recursive Solution. Designing Recursive Functions, Fibonacci Numbers Textbook 1: Chapter 4 (4.1 to 4.6), Chapter 6(6.9) (05 Hours)</p>	
<p style="text-align: center;">Module 4</p> <p>ARRAYS: Concepts, Using Arrays in C, Declaration and Definition, Accessing Elements in Arrays, Storing Values in Arrays, Precedence of Array References, Index Range Checking Inter-Function Communication: Passing Individual Elements, Passing the Whole Array Two-Dimensional Arrays: Declaration, Passing A Two-Dimensional Array STRINGS: C Strings: Storing Strings, The String Delimiter, String Literals, Strings and Characters. Declaring Strings, Initializing Strings, Strings and the Assignment Operator, Reading and Writing Strings String Manipulation Functions: String Length, String Copy, String Concatenate, Textbook 1: Chapter 8 (8.1-8.3, 8.7), Chapter 9 (9.2), Chapter 11 (11.1 to 11.5) (06 Hours)</p>	
<p style="text-align: center;">Module 5</p> <p>POINTERS: Pointer Constants, Pointer Values, Pointer Variables, Accessing Variables Through Pointers , Pointer Declaration and Definition, Declaration versus Redirection, Initialization of Pointer Variables The Type Definition (typedef) Structure: Structure Type Declaration, Initialization, Accessing Structures Operations on Structures, Complex Structures Unions: Referencing Unions, Initializers Textbook 1: Chapter 8(8.5 to 8.6), Chapter 9(9.1), Chapter 12(12.1 to 12.4) (05 Hours)</p>	
<p style="text-align: center;">Experiments</p>	
1.	Compute the roots of a quadratic equation by accepting the coefficients. Print appropriate messages
2.	An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: beyond 300 units Rs 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges
3.	Write a C Program to display the following by reading the number of rows as input, 1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1 1 2 3 2 1 1 2 1 1
4.	Implement Binary Search on Integers.
5.	Implement Matrix multiplication and validate the rules of multiplication.
6.	Compute $\sin(x)/\cos(x)$ using Taylor series approximation. Compare your result with the built-in library function. Print both the results with appropriate inferences.

7.	Using functions sort the given set of N numbers (Bubble sort).
8.	Write user defined functions to implement string operations such as compare, concatenate, and find string length. Use the parameter passing techniques.
9.	Implement structures to read, write and compute average- marks of the students, list the students scoring above and below the average marks for a class of N students.
10.	Develop a program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of N real numbers.
<p>Course Outcomes: The students will be able to:</p> <p>CO1: Define key concepts and terminology related to C programming, including data types, operators, control structures, and functions.</p> <p>CO2: Implement fundamental algorithms and use C programming constructs to solve simple and moderately complex problems.</p> <p>CO3: Examine code to identify and correct logical, syntax, and runtime errors.</p> <p>CO4: Design and develop a functional C program that integrates multiple concepts learned throughout the course, demonstrating an understanding of program structure, data management, and problem-solving techniques.</p> <p>TEXTBOOKS:</p> <p>1. Computer Science : A Structured Programming Approach Using C, by Behrouz A. Forouzan, Richard F. Gilberg, Third Edition, Cengage India Private Limited, ISBN 9788131503638, January 2007.</p> <p>REFERENCE BOOKS:</p> <p>1. Brian W. Kernighan and Dennis M. Ritchie, The 'C' Programming Language, Prentice H all of India..</p> <p>2. Computer fundamentals and programming in c, "Reema Thareja", Oxford University, Second edition, 2017.</p> <p>Alternate Assessment Tools (AATs) suggested:</p> <ul style="list-style-type: none"> • Certification courses on Infosys springboard/NPTEL etc. • Presentation <p>Web links / e – resources:</p> <p>1. E-learning.vtu.ac.in/econtent/courses/video/BS/15PCD23.html</p> <p>2. https://nptel.ac.in/courses/106/105/106105171/ MOOC courses can be adopted for more clarity in understanding the topics and verities of problem solving methods.</p>	