## SEMESTER – I

## PRINCIPLES OF PROGRAMMING USING C (INTEGRATED)

| | | | |
|---|---|---|---|
| **Subject Code** | **23CSE13** | **CIE Marks** | **50** |
| **Hours/Week (L: T: P)** | **3:0:2** | **SEE Marks** | **50** |
| **Total Hours** | **40** | **Examination Hours** | **3** |
| **No. of Credits: 4** | | | |

**Course Learning Objectives:**

The course will enable students to:

| | |
|---|---|
| CLO1 | Understand the steps in the development of a computer program and the structure of a C-language program |
| CLO2 | Understand how decisions are more, design and implement programs with functions |
| CLO3 | Understand basic loop concepts and write programs that use the appropriate loop construct for a given problem |
| CLO4 | Understand the basic concepts and uses of arrays and pointers |

| **CONTENTS** | **# of Hours / RBT Levels** |
|---|---|
| **MODULE 1**<br>**INTRODUCTION TO COMPUTER PROGRAMMING**<br>**Introduction to Computers:** Computing Systems, Computing Environments, Computer Languages, Creating and Running Programs, System Development<br>**Introduction to the C Language:** Structure of a C Program, Identifiers, Types, Variables, Constants, Input/Output, Programming Examples.<br>**Textbook: T1**<br>**Chapters: 1 and 2.** | **08 Hours**<br>**L2** |
| **MODULE 2**<br>**STRUCTURE OF A C PROGRAM AND FUNCTIONS**<br>**Structure of a C Program:** Expressions, Precedence and Associativity, Side Effects, Type Conversion, Statements.<br>**Functions:** Designing Structured Programs, Functions in C, User-Defined Functions.<br>**Textbook: T1**<br>**Chapters: 3 and 4.1 to 4.3.** | **08 Hours**<br>**L2** |
| **MODULE 3**<br>**FUNCTIONS CONTD. AND SELECTION – MAKING DECISIONS**<br>**Functions:** Inter-Function Communication, Standard Functions, Scope, Programming Example – Incremental Development. | **08 Hours**<br>**L3** |

| | |
|---|---|
| **Selection – Making Decisions:** Logical Data and Operators, Two-Way Selection, Multiway Selection, More Standard Functions, Incremental Development Part II.<br>**Textbook: T1**<br>**Chapters: 4.4 to 4.7 and 5.** | |
| **MODULE 4**<br>**REPETITION**<br>**Repetition:** Concept of a loop, Pretest and Post-test Loops, Initialization and Updating, Event- and Counter-Controlled Loops, Loops in C, Loop Examples, Other Statements Related to Looping, Looping Applications, Recursion, Programming Example – The Calculator Program.<br>**Textbook: T1**<br>**Chapter: 6** | **08 Hours**<br>**L3** |
| **MODULE 5**<br>**INTRODUCTION TO ARRAYS AND POINTERS**<br>**Arrays:** Concepts, Using Arrays in C, Inter-Function Communication<br>**Pointers:** Introduction, Pointers for Inter-Function Communication<br>**Textbook: T1**<br>**Chapter: 8.1 to 8.3, 9.1 and 9.2** | **08 Hours**<br>**L3** |

**Laboratory Component**

**List of Experiments**

1. Write a C program to create a customer’s bill for a small restaurant. The restaurant sells only six different items: Idly, Vada, Dosa, Coffee, Tea, and Fruit Juice. The unit prices are Rs. 40, Rs. 20, Rs. 50, Rs. 15, Rs. 15, and Rs. 30, respectively. The program must read the quantity of each item ordered from the keyboard. It then calculates the cost of each item, the subtotal and the total cost after a 5% GST. The input data consist of a set of integers representing the quantities of each item sold. These integers must be input into the program in a user-friendly way.

2. Write a C program to compute the real roots of a quadratic equation (ax2 + bx + c = 0). The program is to prompt the user to enter the constants a, b, and c. It is then to display the roots based on the following rules:
   a. It both a and b is zero, there is no solution.
   b. If a is zero, there is only one root (-c/b).
   c. If the discriminate (b2 – 4ac) is negative, there are no real roots.
   d. For all other combinations, there are two roots.

3. Write a C program that reads a series of numbers and calculates the average, geometric mean, and harmonic mean.

4. Write a C program that determines a student's grade. It reads three test scores (between 0 and 50) and calls a function that calculates and returns a student's grade based on the following rules:
    a. If the average score is 90% or more, the grade is A.
    b. If the average score is 70% or more and less than 90%, it checks the third score. If the third score is more than 90%, the grade is A; otherwise, the grade is B.
    c. If the average score is 50% or more and less than 70%, it checks the average of the second and the third scores. If the average of the two is greater than 70%, the grade is C; otherwise, it is D.
    d. If the average score is less than 50%, then the grade is F.

5. Write a C program that asks the user to enter the current date and a person's date of birth in the form day, month, year. The program then calculates the person's age in integral years. Use separate functions to enter the dates, calculate the person's age, and print the results.

6. Write a C program that computes Sin(x) using Taylor series approximation. Compare the result with the built-in library function.

7. Write a C program to accept the number as a parameter through a user defined function and find its factorial by using recursion.

8. Write a C program that uses a function that tests if every element of array A is equal to its corresponding element in array B. The function is to return true if all elements are equal and false if at least one element is not equal.

9. Write a C program that uses a function that reverses the elements of an array so that the last element becomes the first, the second from the last becomes the second, and so forth.

10. Write a C program that uses pointers to add, subtract, multiply and divide two numbers.

**Course Outcomes:** Upon successful completion of this course, student will be able to

| | |
|---|---|
| **CO13.1** | Understand problem solving steps and construct algorithm/pseudocode/flow chart that can be converted into a program. |
| **CO13.2** | Write appropriate functions to perform repetitive tasks in any application. |
| **CO13.3** | Write programs that need decision making and looping constructs for a given problem. |
| **CO13.4** | Store and retrieve data in arrays. |
| **CO13.5** | Understand the concept and use of pointers in writing programs. |

**Textbooks:**

1. Computer Science - A Structured Programming Approach Using C, Behrouz A. Forouzan and Richard F. Gilberg, 3$^{rd}$ Edition, Cengage Learning India Pvt. Ltd., 2007, ISBN-10:81-315-00363-1

### Reference Books:

1. The C Programming Language, Kernighan B. W. and Dennis M. Ritchie, 2nd Edition, Pearson Education India, 2015, ISBN: 978-93-3254-944-9
2. Let us C, Yashvant P. Kanetkar, 16th Edition, BPB Publications, 2019, ISBN: 978– 93-8728-449-
3. Programming in C, Reema Thareja, 2nd Edition, Oxford University Press, 2015, 978-0-19-9456147

### E-Books / Web References

1. https://en.wikibooks.org/wiki/A_Little_C_Primer

### MOOCs

1. http://lms.vtu.ac.in/econtent/courses/BS/14CPL16/index.php
2. https://nptel.ac.in/courses/106/105/106105171/
3. https://www.coursera.org/learn/c-for-everyone

### Mapping of CO-PO:

| CO/PO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO13.1** | 3 | 3 | 2 | 1 | 1 | | | 1 | 2 | 1 | | 1 | 2 | |
| **CO13.2** | 3 | 3 | 2 | 1 | 1 | | | 1 | 2 | 1 | | 1 | 2 | |
| **CO13.3** | 3 | 3 | 2 | 1 | 1 | | | 1 | 2 | 1 | | 1 | 2 | |
| **CO13.4** | 3 | 3 | 2 | | 1 | | | 1 | 2 | 1 | | 1 | 2 | |
| **CO13.5** | 3 | 3 | 2 | | 1 | | | 1 | 2 | 1 | | 1 | 2 | |
| **Average** | **3** | **3** | **2** | **1** | **1** | | | **1** | **2** | **1** | | **1** | **2** | |

**Low-1: Medium-2: High-3**