Course Title: Computer Programming Lab in C			
Type of Course: DCC (Discipline Centric Core)	**Level of Course:** 4.5	**Delivery Sub Type of the course:** Practical	
Course code: CSE-3-104-P	**No. of credits:** 2	**T-P-S:** 0-4-0	**Learning hours:** 60
Pre-requisite and Co-requisite of Course: No prior programming knowledge required.			
Department: Computer Science and Engineering			

SYLLABUS

Course objectives

- To implement and understand control structures like loops (for, while, do-while) and conditional statements (if, switch).
- To learn to debug and troubleshoot C programs using debugging tools and techniques to identify and fix errors.
- To implement operations to manipulate data structures like arrays and structures.
- To learn and implement efficient algorithms and data structures to optimize code performance and reduce execution time.

Course content

Module /Unit	Topic	T	P	S
1	**Introduction to C Programming** • Write a program to display "Hello, World!" on the screen. • Write a program to add two integers and display the result. • Write a program to calculate the area of a circle. The radius should be taken as user input. • Write a program to determine the roots of a quadratic equation. The values of a, b, and c should be taken as user input. • Write a program to calculate the simple interest and compound interest. • Write a program to convert degree Celsius to Fahrenheit and Fahrenheit to Celsius. • Write a program to find the ASCII value of a character. • Write a program to convert Kilometers to Meters and vice versa. • Write a program to convert the height of a person from inches to centimeters. • Write a program to find the size of int, float and char.	0	10	0
2	**Decision Making** • Write a program to check if a number entered by the user is positive, negative, or zero. • Write a program to check whether a given is a leap year or not. • Write a program to find the largest number among the three numbers • Write a program to calculate grades according to the marks of the student. • Write a program to check whether the input entered by the user is a vowel or consonant. • Write a menu-driven program to perform arithmetic operations (addition, subtraction, multiplication, and division) on two numbers using switch-case.	0	5	0

23/C

3	**Loops** • Write a program to find the sum of natural numbers. • Write a program to find the sum of a geometric series using loops. • Write a program to print the series $1^2+2^2+\ldots+N^2$ • Write a program to display fibonacci series • Write a program to find the factorial of a given number using loops. • Write a program to display characters from A to Z using Loop • Write a program to display armstrong numbers within a range specified by the user	0	5	0
4	**Functions** • Write a function to swap two numbers using pass-by-value. • Write a function to print the first 'n' prime numbers. • Write a recursive program to print the first 'n' Fibonacci numbers. • Write a program to reverse a sentence using recursion • Write a program to convert using decimal to binary using recursion. • Write a program to find the GCD of a number using a function. • Write a program to calculate the power of a number using recursion.	0	6	0
5	**Arrays** • Write a program to find the largest and smallest elements in an array of integers. • Write a program to multiply two 3x3 matrices. • Write a program to find the sum of elements in the array. • Write a program to find the frequency of a number in the array • Write a program to search an element in the array • Write a program to print the transpose of a matrix	0	4	0
6	**Strings** • Write a program to perform the following operators on Strings with and without using String library functions: length of a string, concatenate 2 strings, reverse a string, copy one string to another string. • Write a program to check if two strings are equivalent. • Write a program to calculate the number of words in a sentence.	0	5	0
7	**Pointer** • Write a program to demonstrate the use of pointers by declaring a pointer to an integer, assigning it the address of an integer variable, and displaying the value using the pointer. • Write a C program to swap two numbers using call by reference. • Write a program to increment and decrement a pointer • Write a program to compare two pointers.	0	5	0
8	**Structure and union** • Write a program to define a structure and union for a student with fields for name, roll number, and marks. Input and display the details of a student. • Write a program to create an array of structures for 5 students and find the student with the highest marks.	0	5	0

	• Write a program to create an array of unions for 5 students and find the student with the lowest marks. • Write a program to demonstrate the difference between union and structure.			
9	**File handling** • Write a program to read a text file and display its contents on the screen. • Write a program to write user input data to a file and then read and display the data from the file. • Write a program to copy the contents from one file to another. • Write a program to count the number of words and spaces in a file. • Write a program to replace a specific line with another text in a file.	0	5	0
10	• A real-world application using C of the students' choice , for e.g. Library Management System, Class Record Management System, games, etc.	0	10	0

Scheme of End Semester Examination **Total: 100 marks**

As per DSEU Regulation - 2(A), 2024

Recommended Books and References:

- Herbert Schildt, "C: The Complete Reference", Osbourne Mcgraw Hill, 4th Ed.
- "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie, 2nd Ed.
- "Programming in ANSI C" by E. Balagurusamy, 8th Ed.
- "Let Us C" by Yashavant Kanetkar, 19th Ed.
- Schaum's Outlines "Programming with C" by Byron Gottfried, 2nd Ed.

Learning outcomes

By the end of the course, students will be able to:

- Implement error handling strategies to create robust programs.
- Design and implement logical solutions to problems using appropriate control flow mechanisms
- Use structures and unions to handle complex data types and optimize memory usage in programs.
- Perform basic file operations, including reading from and writing to files, and manage file pointers effectively.

Hyperlinks of suggested e-Resources:

- Coursera - C Programming for Everybody
- NPTEL – Programming in C

Hyperlinks of suggested e-resources on the web

Pedagogical approach

Practical Demonstration: Installation of IDE or use of online editors/compilers, writing, compiling, executing and debugging a program.

Hands-on Practice: Emphasis on writing and executing programs during lab sessions.

Problem-Solving: Practicals designed to develop problem-solving skills.

Peer Learning: Encouraging collaboration and peer-to-peer learning for complex tasks.

Projects: A Small project to integrate and apply all concepts learned.

Additional information (if any)