COMPUTER PROGRAMMING LABORATORY

I Semester: **Common for CSE / ECE / EEE / IT** | **II Semester:** **Common for AE / CE / ME**

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
ACS101	Foundation	-	-	3	2	30	70	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 36				Total Classes: 36		

I. COURSE OVERVIEW:

The course covers the basics of programming and demonstrates fundamental programming techniques, customs and terms including the most common library functions and the usage of the preprocessor. This course helps the students in gaining the knowledge to write simple C language applications, mathematical and engineering problems. This course helps to undertake future courses that assume this programming language as a background in computer programming. Topics include variables, data types, functions, control structures, pointers, strings, arrays and dynamic allocation principles. This course in reached to student by power point presentations, lecture notes, and lab involve the problemsolving in mathematical and engineering areas.

II. OBJECTIVES:

The course should enable the students to:

II The demonstration knowledge of basic abstract data types (ADT) and associated algorithms for organizing programs into modules using criteria that are based on the data structures of the program.

III The practical implementation and usage of non linear data structures for solving problems of different domain.

IV The knowledge of more sophisticated data structures to solve problems involving balanced binary search trees, AVL Trees, B-trees and B+ trees, hashing.

V The graph traversals algorithms to solve real-world challenges such as finding shortest paths on huge maps and assembling genomes from millions of pieces

III. COURSE OUTCOMES:

After successful completion of the course, students should be able to:

CO 1	**Demonstrate** problem solving steps in terms of algorithms, pseudo code and flowcharts for Mathematical and Engineeringproblems. .	Understand
CO 2	**Make use** the concept of operators, precedence of operators, conditional statements and looping statements to solve real timeapplications.	Apply
CO 3	**Demonstrate** the concept of pointers, arrays and performpointer arithmetic, and use the preprocessor's.	Understand
CO 4	**Analyze** the complexity of problems, modularize the problemsinto small modules and then convert them into programs.	Apply
CO 5	**Implement** the programs with concept of file handling functionsand pointer with real time applications of C.	Apply
CO 6	**Explore** the concepts of searching and sorting methods with realtime applications using c	Analyze

IV. SYLLABUS:

LIST OF EXPERIMENTS

Week-1	OPERATORS AND EVALUATION OF EXPRESSIONS

a. Write a C program to check whether a number is even or odd using ternary operator.

b. Write a C program to perform the addition of two numbers without using + operator.

c. Write a C program to evaluate the arithmetic expression ((a + b / c * d - e) * (f - g)). Read the values a, b, c, d, e, f, g from the standard input device.
d. Write a C program to find the sum of individual digits of a 3 digit number.
e. Write a C program to read the values of x and y and print the results of the following expressions in one line:
   i. (x + y) / (x - y)
   ii. (x + y)(x - y)

Week-2 CONTROL STRUCTURES

a. Write a C program to find the sum of individual digits of a positive integer.
b. A Fibonacci sequence is defined as follows: The first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.
c. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
d. A character is entered through keyboard. Write a C program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol using if-else and switch case. The following table shows the range of ASCII values for various characters.

Characters	**ASCII values**
A – Z	65 – 90
a – z	97 – 122
0 – 9	48 – 57
Special symbols	0 – 47, 58 – 64, 91 – 96, 123 – 127

e. If cost price and selling price of an item is input through the keyboard, write a program to determine whether the seller has made profit or incurred loss. Write a C program to determine how much profit or loss incurred in percentage.

Week-3 CONTROL STRUCTURES

a. Write a C program, which takes two integer operands and one operator from the user, performs the operation and then prints the result. (Consider the operators +, -, *, /, % and use switch statement).
b. Write a C program to calculate the following sum:

$$sum = 1 – x^2 /2! + x^4 /4! – x^6 /6! +x^8 /8! – x^{10}/10!$$

c. Write a C program to find the roots of a quadratic equation.
d. Write a C program to check whether a given 3 digit number is Armstrong number or not.
e. Write a C program to print the numbers in triangular form

```
1
1  2
1  2  3
1  2  3   4
```

Week-4 ARRAYS

a. Write a C program to find the second largest integer in a list of integers.
b. Write a C program to perform the following:
   i. Addition of two matrices
   ii. Multiplication of two matrices
c. Write a C program to count and display positive, negative, odd and even numbers in an array.
d. Write a C program to merge two sorted arrays into another array in a sorted order.
e. Write a C program to find the frequency of a particular number in a list of integers.

Week-5 STRINGS

a. Write a C program that uses functions to perform the following operations:
   i. To insert a sub string into a given main string from a given position.
   ii. To delete n characters from a given position in a given string.

b. Write a C program to determine if the given string is a palindrome or not.

c. Write a C program to find a string within a sentence and replace it with another string.

d. Write a C program that reads a line of text and counts all occurrence of a particular word.

e. Write a C program that displays the position or index in the string S where the string T begins, or 1 if S doesn't contain T.

Week-6 FUNCTIONS

a. Write C programs that use both recursive and non-recursive functions
   i. To find the factorial of a given integer.
   ii. To find the greatest common divisor of two given integers.

b. Write C programs that use both recursive and non-recursive functions
   i. To print Fibonacci series.
   ii. To solve towers of Hanoi problem.

c. Write a C program to print the transpose of a given matrix using function.

d. Write a C program that uses a function to reverse a given string.

Week-7 POINTERS

a. Write a C program to concatenate two strings using pointers.

b. Write a C program to find the length of string using pointers.

c. Write a C program to compare two strings using pointers.

d. Write a C program to copy a string from source to destination using pointers.

e. Write a C program to reverse a string using pointers.

Week-8 STRUCTURES AND UNIONS

a. Write a C program that uses functions to perform the following operations:
   i. Reading a complex number
   ii. Writing a complex number
   iii. Addition and subtraction of two complex numbers
   iv. Multiplication of two complex numbers. Note: represent complex number using a structure.

b. Write a C program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.

c. Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.

d. Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

e. Write a C program to define a structure named DOB, which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.

Week-9 ADDITIONAL PROGRAMS

a. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+\ldots\ldots\ldots\ldots+x^n$. For example: if n is 3 and x is 5, then the program computes 1+5+25+125. Print x, n, the sum. Perform error checking. For example, the formula does not make sense for negative exponents – if n is less than 0. Have your program print an error message if n<0, then go back and read in the next pair of numbers of without computing the sum. Are any values of x also illegal? If so, test for them too.

b. 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C program to find the 2's complement of a binary number.
c. Write a C program to convert a Roman numeral to its decimal equivalent. E.g. Roman number CD is equivalent to 400.

Week-10 PREPROCESSOR DIRECTIVES

a. Define a macro with one parameter to compute the volume of a sphere. Write a C program using this macro to compute the volume for spheres of radius 5, 10 and 15 meters.
b. Define a macro that receives an array and the number of elements in the array as arguments. Write a C program for using this macro to print the elements of the array.
c. Write symbolic constants for the binary arithmetic operators +, -, *, and /. Write a C program to illustrate the use of these symbolic constants.

Week-11 FILES

a. Write a C program to display the contents of a file.
b. Write a C program to copy the contents of one file to another.
c. Write a C program to reverse the first n characters in a file, where n is given by the user.
d. Two files DATA1 and DATA2 contain sorted lists of integers. Write a C program to merge the contents of two files into a third file DATA i.e., the contents of the first file followed by those of the second are put in the third file.
e. Write a C program to count the no. of characters present in the file.

Week-12 COMMAND LINE ARGUMENTS

a. Write a C program to read arguments at the command line and display it.
b. Write a C program to read two numbers at the command line and perform arithmetic operations on it.
c. Write a C program to read a file name at the command line and display its contents.

Reference Books:

1. Yashavant Kanetkar, "Let Us C", BPB Publications, New Delhi, 13th Edition, 2012.
2. Oualline Steve, "Practical C Programming", O'Reilly Media, 3rd Edition, 1997.
3. King K N, "C Programming: A Modern Approach", Atlantic Publishers, 2nd Edition, 2015.
4. Kochan Stephen G, "Programming in C – A Complete Introduction to the C Programming Language", Sam's Publishers, 3rd Edition, 2004.
5. Linden Peter V, "Expert C Programming: Deep C Secrets", Pearson India, 1st Edition, 1994

Web References:

1. https://www.sanfoundry.com/c-programming-examples
2. https://www.geeksforgeeks.org/c
3. https://www.cprogramming.com/tutorial/c
4. https://www.cs.princeton.edu

Course Home Page: