INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad – 500043

Information Technology

List of Laboratory Experiments

PROGRAMMING FOR PROBLEM SOLVING USING C LABORATORY								
Course Code	**Category**	Hours / Week			**Credits**	Maximum Marks		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
ACSC04	**Foundation**	0	0	3	1.5	30	70	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 36**				**Total Classes:36**		
Branch: IT	**Semester: I**	**Academic Year: 2021-22**				**Regulation: UG20**		

Course overview: The main emphasis of the course will be on problem solving aspects in through C programming. The students will understand programming language, programming, concepts of loops, reading a set of data, step wise refinements, functions, control structures, arrays, dynamic memory allocations, enumerated data types, structures, unions, and file handling. This course provides adequate knowledge to solve problems in their respective domains.

Course objectives:

The students will try to learn:

I. Problem-solving through programming
II. Programming language, programming, reading a set of Data, stepwise refinement, concepts of Loops, Functions, Control structure, Arrays, Structure, Pointer and File concept.
III. To build efficient programs in 'C' language essential for future programming and software engineering courses.

Course outcomes:

CO 1: Illustrate problem solving steps in terms of algorithms, psuedocode and flowcharts for Mathematical and Engineering problems.
CO 2: Demonstrate basic and derived data types, operators in C program statements.
CO 3: Construct programs involving decision structures, loops, arrays and strings.
CO 4: Make use of various types of functions, parameters, and return values for complex problem solving.
CO 5: Illustrate the dynamic memory management by using pointers.
CO 6: Extend file input and output operations in implementation of real time applications.

WEEK NO	EXPERIMENT NAME	CO

WEEK – I	**OPERATORS AND EVALUATION OF EXPRESSIONS**
	
	 a. Design and develop a flowchart and algorithm to read a number and implement using a C program to check whether the given number is even or odd using ternary operator.
	 b. Design and develop a flowchart and algorithm to read two integers and implement using a C program to perform the addition of two numbers without using +operator.
	 c. Develop a C program to evaluate the following arithmetic expressions by reading appropriate input from the standard input device. Understand the priority of operators while evaluating expressions.
	   i. 6*2/(2+1 * 2/3 +6) +8 * (8/4)
	   ii. 17 – 8 / 4 * 2 + 3 - ++2
	   iii. ! (x > 10) && (y = = 2)
	 d. Develop a C program to display the size of various built-in data types in C language.
		CO 1

WEEK – II	**CONTROL STRUCTURES** a. Design and develop a flowchart and algorithm to read a year as an input and find whether it is leap year or not. Implement a C program for the same and execute for all possible inputs with appropriate messages. Also consider end of the centuries. b. Design and develop a flowchart and algorithm to find the square root of a given number N. Implement a C program for the same and execute for all possible inputs with	**CO 1**

		appropriate messages. (Note: Don't use library function sqrt(n), Hint: Use Newton-Raphson method to find the square root). c. Design and develop a flowchart and algorithm to generate a Fibonacci sequence up to a given number N. 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. Implement a C program for the developed flowchart/algorithm and execute the same to generate the first N terms of the sequence. d. Design and develop a flowchart and algorithm that takes three coefficients (a, b, and c) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. Implement a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages.	
WEEK – III	**CONTROL STRUCTURES**		**CO 3**
	a. Design and develop an algorithm to find the reverse of an integer number N and check whether it is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: N: 2020, Reverse: 0202, Not a Palindrome. b. Draw the flowchart and write C Program to compute sin(x) using Taylor series approximation given by $$\sin(x) = x - (x^3/3!) + (x^5/5!) - (x^7/7!) + \ldots\ldots$$ Compare the result with the built-in Library function and print both the results with appropriate messages. c. Design and develop an algorithm and flowchart to read a three digit number and check whether the given number is Armstrong number or not. Write a C program to implement the same and also display the Armstrong numbers between the ranges 1 to 1000. d. Design and develop an algorithm for evaluating the polynomial $f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0$, for a given value of x and its coefficients using Horner's method. Implement a C program for the same and execute the program for different sets of values of coefficients and x.		
WEEK – IV	**ARRAYS**		**CO 3**
	a. Develop, implement and execute a C program to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers. b. Develop, implement and execute a C program to read a list of integers and store it in a single dimensional array. Write a C program to count and display positive, negative, odd and even numbers in an array. c. Develop, implement and execute a C program to read a list of integers and store it in a single dimensional array. Write a C program to find the frequency of a particular number in a list of integers. d. Develop, implement and execute a C program that reads two matrices A (m x n) and B (p x q) and Compute the product A and B. Read matrix A and matrix B in row major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.		
WEEK – V	**STRINGS**		**CO 3**
	a. Develop a user-defined function **STRCOPY (str1, str2)** to simulate the built-in library function **strcpy (str1, str2)** that copies a string str2 to another string str1. Write a C program that invokes this function to perform string copying. Also perform the same operation using built-in function. b. Develop a user-defined function **STRCONCT (str1, str2)** to simulate the built-in library function **strcat (str1, str2)** that takes two arguments str1 and str2, concatenates str2 and str1 and stores the result in str1. Write a C program that invokes this function to perform string concatenation. Also perform the same operation using built-in function. c. Develop a C program that returns a pointer to the first occurrence of the string in a given string using built-in library function **strstr()**. Example: **strstr()** function is used to locate first occurrence of the string "test" in the string "This is a test string for testing". Pointer is returned at first occurrence of the string "test". d. Develop a C program using the library function **strcmp (str1, str2)** that compares the string pointed to by str1 to the string pointed to by str2 and returns an integer. Display		

		appropriate messages based on the return values of this function as follows −	
		if return value < 0 then it indicates str1 is less than str2.	
		if return value > 0 then it indicates str2 is less than str1.	
		if return value = 0 then it indicates str1 is equal to str2.	
WEEK – VI	**FUNCTIONS**		
	a. Design and develop a recursive and non-recursive function **FACT(num)** to find the factorial of a number, n!, defined by FACT(n) = 1, if n = 0. Otherwise FACT(n) = n * FACT(n-1). Using this function, write a C program to compute the binomial coefficient. Tabulate the results for different values of n and r with suitable messages b. Design and develop a recursive function **GCD (num1, num2)** that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers. c. Design and develop a recursive function **FIBO (num)** that accepts an integer argument. Write a C program that invokes this function to generate the Fibonacci sequence up to num. d. Design and develop a C function **ISPRIME** (num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges. e. Design and develop a function **REVERSE (str)** that accepts a string arguments. Write a C program that invokes this function to find the reverse of a given string.	**CO 3**
WEEK – VII	**POINTERS**		
	a. Develop a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers. b. Develop a C program to read a list of integers and store it in an array. Then read the array elements using a pointer and print the value along with the memory addresses. c. Design and develop non-recursive functions **input_matrix(matrix, rows, col**s) and **print_matrix(matrix, rows, cols)** that stores integers into a two-dimensional array and displays the integers in matrix form. Write a C program to input and print elements of a two dimensional array using pointers and functions. d. Develop a C program to a store a list of integers in a single dimensional array using dynamic memory allocation (limit will be at run time) using malloc() function. Write a C program to read the elements and print the sum of all elements along with the entered elements. Also use free() function to release the memory.	**CO 4**
WEEK –VIII	**STRUCTURES AND UNIONS**		
	a. Write a C program that uses functions to perform the following operations: a. Reading a complex number b. Writing a complex number c. Addition and subtraction 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.	**CO 4**
WEEK - IX	**ADDITIONAL PROGRAMS**		
	a. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+\ldots +x^n$. 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. Develop a C program to find the 2's complement of a given binary number. 2's complement is obtained by scanning it from right to left and complementing all the	**CO 4**

		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. Develop a C program to convert a Roman numeral to its decimal equivalent. E.g. check for the inputs - Roman number IX is equivalent to 9 and Roman number XI is equivalent to 11.	
WEEK - X	**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 15meters. 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.		**CO 5**
WEEK - XI	**FILES**		

	a. Create an employee file **employee.txt** and write 5 records having employee name, designation, salary, branch and city. Develop a C program to display the contents of **employee.txt** file.
	b. Create a **studentolddata.txt** file containing student name, roll no, branch, section, address. Develop a C program to copy the contents of **studentolddata.txt** file to another file **studentnewdata.txt**.
	c. Develop a C program to create a text file **info.txt** to store the information given below. Implement using a C program to count the number of words and characters in the file **info.txt**.
	
	    Test Data:
	
	    Input the file name to be opened : info.txt
	
	    Expected Output:
	
	    The content of the file info.txt are :
	
	    Welcome to IARE
	
	    Welcome to Computer Programming
	
	    The number of words in the file info.txt are : 7
	
	    The number of characters in the file info.txt are : 46
	
	d. Given two university information files "**studentname.txt**" and "**roll_number.txt**" that contains students Name and Roll numbers respectively. Write a C program to create a new file called "**output.txt**" and copy the content of files "**studentname.txt**" and "**roll_number.txt**" into output file. Display the contents of output file "**output.txt**" on to the screen.
	studname.txt **roll_number.txt**
	
	Asha 20951A1201
	
	Bharath 20951A0502
	
	Uma 20951A0456
	
	Shilpa 20951A0305
			CO 5

WEEK - XII	**COMMAND LINE ARGUMENTS**		
	a. Develop a C program to read a set of arguments and display all arguments given through command line. b. Develop a C program to read a file at command line argument and display the contents of the file. c. Develop a C program to read N integers and find the sum of N integer numbers using command line arguments. d. Develop a C program to read three integers and find the largest integer among three using command line argument.		**CO 6**