Mr. Ashok G. Badresiya| A.Y.: 2025-26 | Sem./Year: 2 | Course: 2304CS201 – C Programming - II | Slot Type: Lab (TA) | Division: BCA & BSc IT-2A

Sr.	Course Content	Planning Title	Planning Description
1	Revision of Decision-Making & Loops	Lab – 1	Revision of Decision-Making & Loops 1. W.A.P. to print the multiplication table of 'n' numbers. **(A)** 2. W.A.P. to find out the factorial of a number. **(A)** 3. W.A.P. to find out whether the entered number is prime or not. **(A)** 4. W.A.P. to print the following pattern. **(B)** 1 23 456 78910 5. W.A.P. to read a number from the user and print the reverse of that number. **(B)** 6. Write a program to enter a number and print sum of each digit of the number using a while loop. **(C)**
2	Revision of Arrays and Strings	Lab – 2	Revision of Arrays & Strings 1. W.A.P. to sort the integer numbers in ascending order. **(A)** 2. W.A.P. to add two 3*3 matrices. **(A)** 3. W.A.P. to calculate the length of a string without using the inbuilt function. **(A)** 4. W.A.P. to reverse a string without using the built-in function. **(B)** 5. W.A.P. to copy a string into another string without using the built-in function. **(B)** 6. W.A.P. to perform an insertion operation on 1 – D array. **(C)**
3	Basic Programs Using User-Defined Function	Lab – 3	Basic Programs Using User-Defined Function 1. Add two numbers using a user-defined function. **(A)** 2. Find the maximum number between two numbers using a function. **(A)** 3. Count simple interest using a function. **(A)** 4. Generate a Fibonacci series of N given numbers using the function name fibbo(). **(B)** 5. Find given number is prime or not using a function. **(C)**
4	Intermediate Programs Using User-Defined Function	Lab – 4	Intermediate Programs Using User-Defined Function 1. Return the maximum of three floating-point numbers. **(A)** 2. Swap two numbers using call by value. **(A)** 3. Swap two numbers using call by reference. **(A)** 4. Perform the task of reversing a number and return that number, then check given number is a palindrome or not. **(B)** 5. Find all prime numbers between given interval using functions. **(C)**
5	Advanced Programs Using User-Defined Function	Lab – 5	Advanced Programs Using User-Defined Function 1. Find the factorial of a number using a function and a recursive function. **(A)** 2. Pass an array in a function to print the array elements. **(A)** 3. Find the power of any number using recursion. **(A)** 4. Swap elements of two integer arrays using user user-defined function. **(B)** 5. Find the reverse of any number using recursion. **(C)**

6	Basic Programs Using Structure and Union	Lab – 6	Basic Programs Using Structure and Union 1. Create, declare and initialise the structure employee. **(A)** 2. Create a structure book with book title, author, publication, and price. Read the data of 3 books and display. **(A)** 3. Demonstrate the difference between structure and union. **(A)** 4. Demonstrate nested structure. **(B)** 5. Add two distances in feet and inches using structure. **(C)**
7	Intermediate Programs Using Structure and Union	Lab – 7	Intermediate Programs Using Structure and Union 1. Create structure student with name, percentage and age. Read data of 5 students using array of structure. **(A)** 2. Create structure student with name, percentage and age. Read data of N students using array of structure. Print details of student with maximum percentage. **(A)** 3. Create structure student with name, percentage and age. Read data of N students using array of structure. Arrange student data alphabetically and print all data. **(B)** 4. Create a structure student with roll number, name and marks of 3 subjects. Read data of N students using array of structure. Calculate total and percentage for each student. Print the details of the top 3 students based on highest total marks. Also print class average percentage. **(C)**
8	Advanced Programs Using Structure and Union	Lab – 8	Advanced Programs Using Structure and Union 1. Define a structure called Players which describes details like p_name, team, score, average. Write a program to read data for 3 players and print that data. **(A)** 2. Define a union called Players which describes details like p_name, team, score, average. Write a program to read data for 3 players and print that data. **(A)** 3. Define a structure called Players which describes details like p_name, team, score, average. Write a program to read data for N players and arrange p_name in alphabetical order. **(B)** 4. Define a structure called Players which describes details like p_name, team, score and average. Write a program to read data for N players using array of structure. Calculate the highest score and display the details of the top 3 players based on score. Also arrange all players in alphabetical order of p_name and print the sorted list. **(C)**
9	Basic Programs Using Pointer	Lab – 9	Basic Programs Using Pointer 1. Print value and address of a variable. **(A)** 2. Demonstrate int, float, double and char pointer. **(A)** 3. Calculate sum of two numbers using pointer. **(A)** 4. Copy one array to another using pointers. **(B)** 5. Swap two arrays using pointers. **(C)**
10	Intermediate Programs Using Pointer	Lab – 10	Intermediate Programs Using Pointer 1. Swap the values of two numbers using a pointer. **(A)** 2. Store n elements in an array and print the elements using pointer.**(A)** 3. Find even and odd numbers in array using pointer. **(A)** 4. Print positive and negative numbers in array using pointers. **(B)** 5. Sort array using pointers. **(C)**

11	Advanced Programs Using Pointer	Lab – 11	Advanced Programs Using Pointer 1. Add two matrices using Pointers. **(A)** 2. Print the sum of columns of a matrix using Pointers. **(A)** 3. Find the length of the string using Pointers. **(A)** 4. Copy one string into another string using Pointers. **(B)** 5. Find Maximum Element in a 2-D Array Using Pointers. **(C)**
12	Programs Using Dynamic Memory Allocation	Lab – 12	Programs Using Dynamic Memory Allocation 1. Allocate and de-allocate memory for int, char and float variables at run time. **(A)** 2. Calculate the sum of n numbers entered by the user using malloc (). **(A)** 3. Calculate the sum of n numbers entered by the user using calloc (). **(A)** 4. Write a program that demonstrates the use of realloc (). **(B)** 5. Allocate dynamic memory for a structure variable. **(C)**
13	Programs Using Pre-processor Directives	Lab – 13	Programs Using Pre-processor Directives 1. Illustrate the use of #define preprocessor. **(A)** 2. Get current time using __TIME__. **(A)** 3. Define a function like Macro that should use printf. Define Macro PRINT to print the given integer argument. **(A)**
14	Programs Using File Handling	Lab – 14	Programs Using File Handling 1. Create, open and close a file. **(A)** 2. Demonstrate functions fprintf(), fscanf(), fputc(), fgetc() **(A)** 3. Count chars, spaces, tabs and new lines in a file. **(A)** 4. Demonstrate functions fseek() and rewind (). **(B)** 5. Copy one file to another file. **(C)**
15	Programs Using Command Line Arguments	Lab – 15	Programs Using Command Line Arguments 1. Perform the addition of two numbers. **(A)** 2. Find the maximum number from two numbers. **(A)** 3. Find the Sum and Average of 5 Numbers. **(A)** 4. Count Even and Odd. **(B)** 5. Count Vowels in a String. **(C)**