Program	Diploma (DE)	**Semester - 1**
Type of Course	Professional Core	
Prerequisite	Basic Knowledge of Computer.	
Course Objective	This hands-on C programming course provides a comprehensive introduction to the ANSI C language. It is an introductory course and covers the key features of the C language and its usage.	

Teaching Scheme (Contact Hours)				Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
Lecture	Tutorial	Lab	Credit	SEE (T)	CIA (T)	SEE (P)	CIA (P)	
3	0	4	5	40	30	20	10	100

***SEE** - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)*

Course Content

T - Teaching Hours | **W** - Weightage

Sr.	Topics	T	W
1	**Introduction** Introduction to flowchart : symbol and usage of flowchart, examples of flowchart, introduction to algorithm : characteristics and usage of algorithm, examples of algorithm, limitations of algorithm and flowchart, history and importance of C, header files, C tokens, keywords and identifiers, data-types, type conversion, constant variable, ASCII codes, structure of C program	10	20
2	**C Operators & Decision Making Statements** Operators in C (arithmetic operator, assignment operator, relational operator, increment/decrement operator, logical operator, bitwise operator, conditional operator, special operator), operator precedence and associativity, simple if statement, if..else statement, nested if...else and else...if ladder, conditional operator, switch statement, break statement	9	20
3	**C Loops** Introduction to loops, while loop, for loop, do...while loop, nested loop, goto statement, continue statement	9	20
4	**Introduction to array** Introduction to array, types of array, declaration, initialization and access of 1-D array, insertion, deletion, merging, searching and sorting operations on 1-D array	9	20
5	**2-D array and String** Declaration, initialization and access of 2-D array, addition, multiplication and transpose of matrix (represented as 2-D array), introduction to string, declaration and initialization of string, string handling functions (strlen, strcpy, strcat, strcmp, strlwr, strupr, strrev), user defined implementation of string handling functions	8	20
	Total	**45**	**100**

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	25	55	20	0	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:	
CO1	**prepare** flowchart and algorithm for real life problem.
CO2	**solve** problems using operators and decision making statements.
CO3	**practice** problems using looping and control statement.
CO4	**perform** array operation on 1 – D array using C programming language.
CO5	**demonstrate** usage of 2 – D array, string handling function in C programming language.

Reference Books

1.	**Programming in ANSI C** By Balagurusamy,E | Tata Mac-Graw Hill Education Private Limited
2.	**Let Us C** By Kanetkar,Yashavant P | BPB Publications
3.	**Mastering C** By Venugopal,k r | Tata Mac-Graw Hill Education Private Limited
4.	**The C Programming Language** By Kernighan Brain W; Ritchie Dennis M | Pearson Education | 2nd Edition

List of Practical

1.	**Basic Algorithm and Flowchart** 1. Calculate addition of two numbers. (A) 2. Calculate average of three numbers. (A) 3. Find area of circle. (pie*r*r) (A) 4. Find area of triangle. ((height*base)/2) (A) 5. Calculate simple interest. (principal*roi*time period)/100 (A) 6. Convert temperature from Fahrenheit to Celsius. (Formula: c=(((f-32)*5))/9) (A)
2.	**Intermediate Algorithm and Flowchart** 1. Convert given feet into inches. (feet*12) (A) 2. Swap two numbers. (Using temporary variable and without using temporary variable) (A) 3. Check whether given number is positive or negative. (A) 4. Check whether the given number is odd or even. (A) 5. Find out largest number from given two numbers. (A) 6. Find out largest number from given three numbers. (B)
3.	**Advanced Algorithm and Flowchart** 1. Display day name for the given number. (A) 2. Accept three numbers from user and print them in ascending and descending order. (A) 3. Check whether the given year is leap year or not. [If a year can be divisible by 4 but not divisible by 100 then it is leap year but if it is divisible by 400 then it is leap year]. (A) 4. Input electricity unit charge and calculate the total electricity bill according to the given condition: • For first 50 units Rs. 0.50/unit • For next 100 units Rs. 0.75/unit • For next 100 units Rs. 1.20/unit • For unit above 250 Rs. 1.50/unit • An additional surcharge of 20% is added to the bill. (B)
4.	**Advanced Algorithm and Flowchart with Loop** 1. Find factorial of given number. (A) 2. Print all factors of given number. (A)

	3. Print fibonacci series upto given numbers. (A) 4. Print sum of digit of given number. (A) 5. Check whether given number is prime or not. (B) 6. Check whether given number is palindrome or not. (C)
5.	**Programs Using Basic Syntax in C** 1. Print "Hello World". (A) 2. Print your address i) using single printf ii) using multiple printf. (A) 3. Print addition of 2 numbers. (with & without scanf) (A) 4. Print average of three numbers. (with & without scanf) (A) 5. Print area of circle. (pie*r*r) (A)
6.	**Program with basic syntax and formula in C** 1. Print area of triangle. ((height*base)/2) (A) 2. Print simple interest. (principal*roi*time period)/100 (A) 3. Print temperature from Fahrenheit to Celsius. (Formula: c=(((f-32)*5))/9) (A) 4. Convert seconds into hours, minutes & seconds and print in HH:MM:SS. [e.g. 10000 seconds = 02:46:40)] (B) 5. Convert number of days into year, week & days. [e.g. 375 days mean 1 year, 1 week and 3 days] (B)
7.	**Programs Using Decision making statement.** 1. Print given feet into inches. (inches = feet*12) (A) 2. Swap two numbers. (Using temporary variable and without using temporary variable) (A) 3. Check whether given number is positive or negative. (A) 4. Check whether the given number is odd or even. (A) 5. Find out largest number from given three numbers. (A) 6. Check whether given character is vowel or consonant. (Using single if only) (B)
8.	**Programs Using Nested if and else if ladder** 1. Perform Addition, Subtraction, Multiplication and Division of 2 numbers as per user's choice. (A) 2. Enter basic salary of an employee and calculate Gross salary according to given conditions: (A) ◦ Basic Salary >= 10000 : HRA = 20% of basic, DA = 80% of basic ◦ Basic Salary >= 20000 : HRA = 25% of basic, DA = 90% of basic ◦ Basic Salary >= 30000 : HRA = 30% of basic, DA = 95% of basic 3. Check whether the entered character is upper case, lower case, digit or any special character. (A) 4. Input an integer number and check the last digit of number is even or odd. (A) 5. Read marks of five subjects. Calculate percentage and print class accordingly. Fail below 35, Pass Class between 36 to 45, Second Class between 46 to 60, First Class between 61 to 70, Distinction if more than 70. (B)
9.	**Advanced programs Using Nested if and else if ladder** 1. Print weekday based on given number. (A) 2. Print month name based on given number. (A) 3. Input electricity unit charge and calculate the total electricity bill according to the given condition: (A) ◦ For first 50 units Rs. 0.50/unit ◦ For next 100 units Rs. 0.75/unit ◦ For next 100 units Rs. 1.20/unit ◦ For unit above 250 Rs. 1.50/unit ◦ An additional surcharge of 20% is added to the bill. 4. Determine the roots of the equation ax2+bx+c=0. (B) 5. Three sides of a triangle are entered through the keyboard, WAP to check whether the triangle is isosceles, equilateral, scalene or right angled triangle. (B) 6. Find the second largest number among three user input numbers. (C)
10.	**Programs Using Switch case and conditional statement** 1. Print day name of week using switch. (A) 2. Print number of days in a month using switch. (A)

	3. Perform Addition, Subtraction, Multiplication and Division of 2 numbers as per user's choice using switch. (A) 4. Read 3 numbers, multiply largest number from first two numbers to third one using switch. (B) 5. Check whether character is an alphabet or not using conditional operator. (B) 6. Print number of days in a month considering leap year using switch. (C)
11.	**Advanced programs Using Switch case and conditional statement** 1. Demonstrate the behaviour of switch case without break. (A) 2. Check whether given number is positive or negative using conditional operator. (A) 3. Find out largest number from given 3 numbers using conditional operator. (A) 4. Check whether number is even number or odd number using conditional operator. (A)
12.	**Basic Programs Using While Loop** 1. Print 1 to 10 using While loop. (A) 2. Print 1 to N using While loop. (A) 3. Print odd numbers between 1 to N. (A) 4. Print numbers between two given numbers which is divisible by 2. (A) 5. Print number and its square root for 0 to 9. (B) 6. Calculate the square of integers 1 through 10. (C)
13.	**Intermediate Programs Using While Loop** 1. Print sum of 1 to n numbers. (A) 2. Get 10 numbers from user print count of odd, even numbers. (A) 3. Print all integer greater than 100 and less than 200 that are divisible by 7 but not divisible by 5. (A) 4. Print first 50 numbers in series 1, 4, 7, 10… (B) 5. Print sum of series 1 + 4 + 9 + 16 + 25 + 36 + …n. (C)
14.	**Advanced Programs Using While Loop** 1. Print sum of series 1 – 2 + 3 – 4 + 5 – 6 + 7 … n. (A) 2. Print multiplication table of a given number. (A) 3. Calculate xy without using power function. (A) 4. Find factorial of the given number. (A) 5. Find factors of the given number. (B) 6. Print all uppercase and lowercase alphabets. (B) 7. Convert given number in words. (i.e. n=3456 output: Three Four Five Six) (C) 8. Convert decimal number to binary. (i.e. n=11 output: 1101) (C)
15.	**Complex programs Using While Loop** 1. Find out sum of first and last digit of a given number. (A) 2. Find the sum and average of different numbers which are accepted by user as many as user wants. (A) 3. Find whether the given number is prime or not. (A) 4. Print digits of given number. (A) 5. Print given number in reverse order. (A) 6. Check whether the given number is perfect or not. (B) 7. Find whether the given number is prime or not using flag. (B) 8. Check whether the given number is palindrome or not. (C) 9. Check whether the given number is Armstrong or not. (C)
16.	**Programs using For Loop** 1. Print 1 to 10 then modify program print 1 to n. (A) 2. Print sum of 1 to n numbers. (A) 3. Print multiplication table of a given number. (A) 4. Calculate xy without using power function. (A) 5. Find factorial of the given number. (A) 6. Print all factors of the given number. (B) 7. Print the Fibonacci Series. (B)

	8. Count frequency of digits in an integer. (C) 9. Print all ASCII character with their values. (C)
17.	**Basic Programs using Do While Loop** 1. Print sum of 1 to n numbers. (A) 2. Get 10 numbers from user print count of odd, even numbers. (A) 3. Print all integer greater than 100 and less than 200 that are divisible by 7 but not divisible by 5. (A) 4. Print sum of series 1 – 2 + 3 – 4 + 5 – 6 + 7 … n. (A) 5. Print multiplication table of a given number. (B) 6. Print first 50 numbers in series 1, 4, 7, 10… (B) 7. Print sum of series 1 + 4 + 9 + 16 + 25 + 36 + …n. (C)
18.	**Advanced programs using Do While Loop** 1. Find out sum of first and last digit of a given number. (A) 2. Find the sum and average of different numbers which are accepted by user as many as user wants. (A) 3. Find whether the given number is prime or not. (A) 4. Print digits of given number. (A) 5. Check whether the given number is perfect or not. (B) 6. Find whether the given number is prime or not using flag. (B) 7. Check whether the given number is palindrome or not. (C) 8. Check whether the given number is Armstrong or not. (C)
19.	**Basic Patterns Programs using Nested For loop** 1. Print following patterns (A) * ** *** **** ***** 1 12 123 1234 12345 5 54 543 5432 54321
20.	**Advanced Pattern Programs Using Nested For Loop** 1. Print following patterns (A) ***** **** *** ** * 12345 1234 123 12

	1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
21.	**Basic Programs Using 1D Array** 1. Read n numbers from user and print in normal and reverse order. (A) 2. Count number of positive or negative number from an array of n numbers. (A) 3. Count number of even or odd number from an array of n numbers. (A) 4. Count numbers higher than the average of an array. (B) 5. Calculate the average, geometric and harmonic mean of n elements in an array. (C)
22.	**Intermediate Programs Using 1D Array** 1. Find Max, Min, Sum, Avg. of given numbers from an array. (A) 2. Read five person height and weight and count the number of person having height greater than 170 and weight less than 50. (A) 3. Sort elements of an array in an ascending order. (A) 4. Count total duplicate elements in an array. (B) 5. Find missing numbers of sequence using array.(in a sequence 1,2,4,5,7,8,10, Missing numbers are 3,6,9) (C)
23.	**Intermediate Programs related to different array operation Using 1D Array** 1. Copy all elements of one array to another. (A) 2. Count total number of negative elements in array. (A) 3. Count number of elements divisible by 3 in array. (A) 4. Delete all duplicate elements from an array. (B) 5. Reverse elements of an array without using second array. (B) 6. Find two largest elements in a one dimensional array. (C)
24.	**Advanced Programs Using 1D Array** 1. Search element in array. (A) 2. Find maximum and minimum value from array. (A) 3. Input a string in character array and print string and length of string. (A) 4. Swap first element with last, second to second last and so on. (B) 5. Insert new value in the sorted array. (C)
25.	**Basic Programs Using Built-In String Functions** 1. Use string handling functions strlen(), strcmp(), strcpy(), strcat(), strrev(), strlwr() and strupr() (A) 2. Print all character of given string. (A) 3. Find a character from given string. (A) 4. Replace a character in given string. (A)
26.	**Advanced Programs Using Built-In String Functions** 1. Find length of given string without using built-in function. (A) 2. Convert a string into upper case and lower case string without using built-in function. (A)

	3. Copy given string into another string without using built-in function. (A) 4. Read two string from user and merge second string into first string without using built-in function. (B) 5. Count occurrence of a given character in a string. (B)
27.	**Basic Programs Using 2D Array** 1. Read values in two-dimensional array and print them in matrix form. (A) 2. Count number of positive, negative and zero elements from 3 X 3 matrix. (A) 3. Read and store the roll no and marks of 20 students using 2D array. (A) 4. Print Transpose of a matrix. (B) 5. Perform Multiplication of two matrices. (B)
28.	**Advanced Programs Using 2D Array** 1. Count number of odd and even elements from N X N matrix. (A) 2. Check given two matrices are identical or not. (A) 3. Read a matrix and print diagonal elements and its sum. (B) 4. Check a given matrix is a sparse matrix or not. (B) 5. Print the upper triangular matrix. (C)

Useful Links
https://www.geeksforgeeks.org/c-programming-language/ https://www.javatpoint.com/c-programming-language-tutorial https://www.tutorialspoint.com/cprogramming/index.htm