GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Diploma Engineering
Level: Diploma
Branch: Information Technology
Subject Code: DI05016061
Subject Name: Structured Programming with C

w. e. f. Academic Year:	2026-27
Semester:	5th
Category of the Course:	PEC-04

Prerequisite:	Basic Computer Skills Logical Thinking and Problem-Solving Abilities
Rationale:	In today's digital era, digitization and automation connect gadgets, home appliances, and even the human body. The key to these connections is programming. Students need to learn the basics of computer programming. This introductory Computer Programming Course focuses on developing logical thinking and programming skills using the C language. These skills can be applied to scientific, research, and business purposes.

Course Outcome:

After Completion of the Course, the student will be able to:

No	Course Outcomes	RBT Level
01	CO1: Write C programs to input and output data in the prescribed formats with/without control structures.	U
02	CO2: Design C programs using arrays and pointers.	A
03	CO3: Implement user-defined functions.	A
04	CO4: Create user-defined data datatypes like a structure and a union	A
05	CO5: Execute file and I/O operations in C.	A

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE(E)	PA(M)	PA(I)	ESE(V)	
3	0	2	4	70	30	20	30	150

Course Content:

Unit No.	Content	No. of Hours	% of Weightage
1.	**Introduction to C Language** 1.1. Art of Programming with Flowchart - Introduction, Basic Notation, Examples, Advantages & Disadvantages 1.2. C language introduction 1.2.1 History, Importance of C 1.2.2 Basic structure of C Program 1.3. Process of compiling and executing a C Program 1.4. Header files, Library functions - Introduction 1.5. Control Statements in C: if-else condition, switch-case statement, loops: entry-controlled and exit-controlled. 1.6. Storage Classes: Automatic Variables, External, Static, Register 1.7. Type Qualifiers: const, volatile, restrict	05	12
2.	**Arrays and Pointers** 2.1. Arrays: 1D Array, Multi-dimensional array, 2.2. Variable Length Arrays, Flexible Array Members, 2.3. Advanced Control Flow: The goto statement, the NULL statement, the comma operator, setjmp and longjmp functions 2.4. Pointers: Initialize, Dereference, Size, Special Types of Pointers (NULL Pointer, Void Pointer, Wild Pointers, Dangling Pointer), 2.5. Advanced Pointers: Double pointers (pointer to a pointer), Function pointers, Pointer to Function 2.6. Multilevel Pointers	09	20
3.	**Functions** 3.1. Need for functions, Definition, Elements of User-defined Functions, 3.2. Function Declaration, Return Values and their Types, Function Calls 3.3. Category of Functions 3.3.1. No Arguments and no Return Values 3.3.2. Arguments but no Return values 3.3.3. Arguments with Return Values 3.3.4. No Arguments but Returns a Value 3.4. Passing Arrays to Functions, Recursion, The Scope, Visibility, and Lifetime of Variables. 3.5. Advanced Function Concepts: Variadic Functions, va_copy, and Inline Functions, _Noreturn Functions	12	28
4.	**User-defined Datatypes**	12	25

	4.1. User-defined datatypes 4.1.1. Unions, Defining a Union 4.1.2. Accessing Union Members 4.1.3. C Structures, Basic Operations of Structure - Access Structure Members, Initialize Structure Members 4.1.4. Copy Structure, Passing Structure to Functions 4.1.5. typedef for Structures, Structure Pointer 4.2. The Preprocessor: Overview, Conditional Compilation 4.3. Include guards and #undef, #pragma and #error 4.4. Macros: Overview, Macros vs. Functions 4.5. Preprocessor Operators, Predefined Macros, Creating your own Macros Advanced Data Types: The #define statement (constants), Using typedef, Complex number types, Designated Initializers		
5.	File Handling in C 5.1. Create a File, Naming a File, Opening a File, Read from File, Write to a File, Close a File 5.2. File Opening Modes, Accessing File Pointer, 5.3. Read and Write in a Binary File, File error handling	07	15
	Total	**45**	**100**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	**32**	**28**	**-**	**-**	**-**

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

Sr. No.	Title of Book	Author	Publication with place, year and ISBN
1	Programming in ANSI C	E. Balagurusamy	McGraw Hills Education, New Delhi; 2019; ISBN: 9789351343202
2	Let us 'C'	Yashavant Kanetkar	BPB Publication, New Delhi; 2020; ISBN: 978-9389845686
3	Programming with ANSI and Turbo C	Ashok N. Kamthane	Pearson Education, New Delhi; 2008; ISBN: 978-8131704370

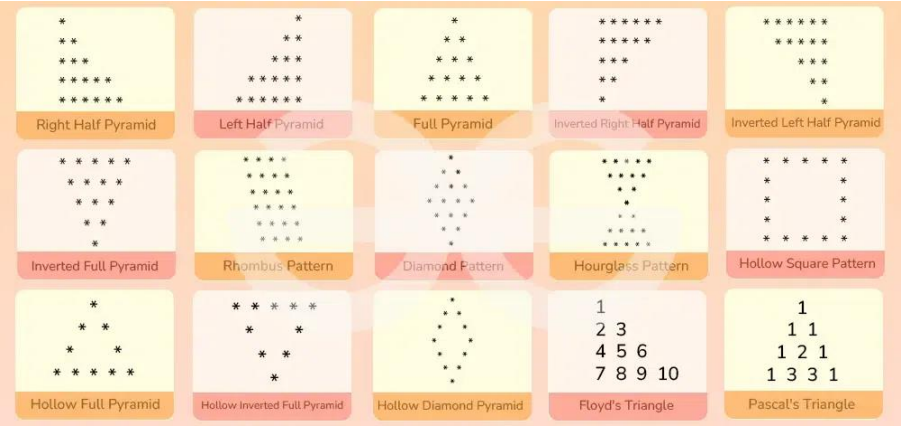
4	Introduction to C Programming	Reema Thareja	Oxford University Press, New Delhi; 2018; ISBN: 9780199492282
5	C in Depth	Srivastava Deepali	BPB Publications, Third Edition, 2022 ISBN: 9788183330480

(b) Open-source software and website:

1. C Compiler for windows/linux
2. https://www.programiz.com/c-programming
3. Online DB : GDB online Debugger | Compiler - Code, Compile, Run, Debug online C, C++ (onlinegdb.com)
4. https://www.youtube.com/watch?v=ywg7cW0Txs4

Suggested Course Practical List:

The following practical outcomes (PrOs) are the subcomponents of the COs. These PrOs need to be attained to achieve the COs.

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. required
1	Write a simple C program to print Hello World.	1	2
2	Write C program using only printf to display below patterns 	1	2
3	Design a program to: Store marks of N students using an array, Use pointers to calculate: Average marks, Highest & lowest marks	2	2

4	Create a program to: Perform addition, subtraction, and multiplication of matrices - Use 2D arrays and pointer arithmetic	2	2
5	Use pointers and malloc() to store employee salaries Calculate: Total salary expense and the highest salary	2	2
6	Create a program using: Pointer and Pointer to pointer. Modify variable values through multiple levels	2	2
7	Create a menu-driven calculator using: Separate functions for: Add, subtract, multiply, divide	3	2
8	Implement factorial and Fibonacci using recursion Compare the recursive vs iterative approach	3	2
9	Pass an array to a function Perform: Sum, Average, and Count even/odd numbers	3	2
10	Create a function that accepts a variable number of arguments, Compute sum/average	3	2
11	Use a struct to store: Name, Roll No, Marks Perform: Add, display, search	4	2
12	Create a union for storing: int, float, char Show memory sharing behavior	4	2
13	Create: Macro for square calculation, Function for square Compare execution	4	2
14	Store student records in file. Perform: Write, Read, and Append	5	2
15	Store employee data (as a struct) using a binary file. Perform: Add record, Search record, and Modify record	5	2
	Total		30

Note:- More Practical Exercises can be designed and offered by the respective course teacher to develop the industry-relevant skills/outcomes to match the COs. The above table is only a suggestive list.

List of Laboratory/Learning Resources Required:

Sr. No.	Laboratory/Learning Resources/Equipment Name with Broad Specifications	PrO. No.
1	Computer with basic configuration with windows	All (1 to 15)
2	C Compilers	All (1 to 15)

Sample Projects: (if required)

1) **Mini Banking System**
   Use:
   - Structures (Account details)
   - Functions (deposit, withdraw)
   - Arrays (multiple users)
   - File handling (persistent storage)
   - Features:
     - Create account
     - Deposit/Withdraw
     - Balance check

2) **Hospital Patient Management System**
   Problem Statement: Design a system to manage hospital patient records with the following features:
   Requirements:
   a) Use structure to store: Patient ID, Name, Age, Disease, Admission Date
   b) Store multiple patients using array + pointers
   c) Use functions for:
      i) Add patient
      ii) Search patient by ID
      iii) Display all patients
      iv) Count patients by disease
   d) Save and retrieve data using file handling
   e) Implement update/delete functionality in file

3) **Inventory Management System for Shop**
   Problem Statement: Create an inventory system to manage products in a shop.

Requirements:

a) Use structure: Product ID, Name, Price, Quantity
b) Store products dynamically using pointers (malloc)
c) Implement functions:
    i) Add new product
    ii) Update stock
    iii) Delete product
    iv) Calculate total inventory value
d) Use file handling to:
    i) Save inventory data
    ii) Load data on program start
e) Use pointer to structure for processing

4) Online Examination Result Processing System

Problem Statement: Develop a system to process and manage student exam results.
Requirements:

a) Use structure: Roll No, Name, Marks of 5 subjects
b) Use arrays to store multiple students
c) Use functions to:
    i) Calculate total, percentage, grade
    ii) Find topper
    iii) Display failed students
    iv) Use pointers to traverse records
d) Store results in file and generate report

Suggested Activities for Students: (if required)

a) Design an algorithm and construct a flowchart for at least 6 problems.
b) Undertake micro-projects in teams.
c) Prepare charts to explain the use/process of the identified topic.
d) Students are encouraged to register themselves in various MOOCs such as: Swayam, edX,
e) Coursera, Udemy etc to further enhance their learning.
f) Encourage students to participate in different coding competitions like Hackathon, online competitions on CodeChef etc.

* * * * * *