25CE101: Fundamentals of Programming

w. e. f. Academic Year:	**2025-26**
Semester:	**1**
Category of the Course:	**Engineering Science**
Prerequisite:	Zeal to learn the subject
Rationale:	The course introduces core programming concepts using C, a foundational language with strong system-level capabilities. It develops logical thinking, problem-solving, and coding skills essential for advanced topics like data structures and operating systems. C's industry relevance and emphasis on efficient coding practices make it ideal for engineering students.

Course Outcomes:

After Completion of the Course, Student will able to:

	Course Outcome (CO)	**RBT Level (Cognitive Domain)**
CO1	Understand basic approaches and strategies used in computational problem solving.	Understand
CO2	Recall and apply flowcharts and algorithms to represent solutions to simple problems.	Remember, Understand
CO3	Formulate logical solutions using flowcharts and algorithms for well-defined problems.	Apply, Analyze
CO4	Understand core concepts and terminology in the C programming language.	Understand
CO5	Analyze program logic and apply control structures to implement modular and efficient code in C.	Understand, Analyze
CO6	Develop, test, and debug C programs using loops, functions, arrays, and decision-making constructs.	Apply, Create

Teaching and Evaluation Scheme:

Teaching Scheme					**Examination Scheme**				
L	**T**	**P**	**C**	**Hrs/Week**	**IE**	**Theory**	**CIA**	**Practical**	**Total Marks**
2	-	4	4	6	40	60	30	20	150

IE: Internal Evaluation
Theory: Theory Exam (End Semester)
CIA: Continuous Internal Assessment
Practical: Practical Exam (End Semester)

Detailed Syllabus:

Topic		Hrs.	% of Weightage
UNIT: 1	**Introduction to Computers and Programming Concepts**	**04**	**12**
Basic block diagram and functions of computer components, Hardware vs Software, Types of software, System software vs Application software, Compiler vs Interpreter, Machine-level, Assembly-level, and High-level programming concepts, Introduction to Flowcharts and Algorithms.			
UNIT: 2	**Introduction to C Language and Program Structure**	**05**	**17**
Features of C, Structure of a C program, Header files, Comments, Data types, Constants and Variables, Operators and Expressions, Evaluation, Type conversion, Precedence rules.			
UNIT: 3	**Control Structures in C**	**06**	**20**
Decision-making structures (if, if-else, nested if), switch-case, Looping (for, while, do-while), Nesting of loops, break, continue, goto statements.			
UNIT: 4	**Arrays and Strings**	**05**	**17**
Concept of Arrays, Declaration and Initialization, One-dimensional and Two-dimensional arrays, String storage and manipulation, Built-in string functions.			
UNIT: 5	**Functions in C**	**05**	**17**
Concept of functions, Function declaration and definition, Calling and returning values, Parameter passing (call by value), Recursion, Preprocessors and Macros.			
UNIT: 6	**Structures and Unions**	**05**	**17**
Definition and declaration of structures, Accessing members, Nested structures, Array of structures, Concept of Unions, Difference between structure and union.			
		30	**100**

List of Practical:

Aim	Hrs
Write a program to print your address.	01
Write a program to perform average of five variables.	01
Write a program to print area of circle, rectangle and square.	01
Write a program to convert years into minutes.	01
Write a program to perform all the arithmetic operations together in a single program.	01
Write a program to print a character entered by user.	01
Write a program to convert small letter case to upper letter case.	01
Write a program to swap the values of two variables using third variable	01
Write a program to swap the values of two variables without using third variable	01
Write a program to find maximum and minimum numbers from two numbers by using Conditional operator.	01
Write a program to demonstrate bitwise operator.	01
Write a program to check whether the entered number is odd or even by using if else statement.	01

Write a program to check whether entered character is alphabet, digit or special symbol.	01
Write a program to find whether entered year is leap year or not.	01
Write a program to check how many days are there in entered month by using switch case.	01
Write a program to check whether entered character is vowel or consonant by using switch statement.	01
Write a program to get maximum number among three.	01
Write a program to calculate grade of given marks.	01
Write a program to print first 10 integers by using go to statement.	01
Write a program to print addition of first n numbers by using go to statement.	01
Write a program to find reverse of given numbers. (Example 132-231)	01
Write a program to check whether entered number is Armstrong or not.	01
Write a program to check whether entered number is palindrome or not.	01
Write a program to print factorial of a given number.	01
Write a program to check whether entered number is prime or not.	01
Write a program to print Different pattern using For Loop.	01
Write a program to print 1 to 5 numbers using array.	01
Write a program to print 1 to 5 reverse numbers using array.	01
Write a program to find sum and average of five numbers.	01
Write a program to find maximum and minimum number from given array.	01
Write a program to find number of positive, negative and zero from given array.	01
Write a program to find number of odd and even from given array.	01
Write a program to sort given n number using array.	01
Write a program to read matrix, display original and transpose of matrix.	01
Write a program to copy one string to another string.	01
Write a program to concatenate two strings.	01
Write a program to find length of given string.	01
Write a program to find length of given string without using string function.	01
Write a program to copy one string to another string without using string function.	01
Write a program to compare two strings.	01
Write a program to reverse a given string.	01
Write a program to find given string is palindrome or not.	01
Write a program to convert a given string into upper case string.	01
Write a user defined function (UDF) to print whether entered number is odd or even.	01
Write a program to add first n numbers using user defined function (UDF).	01
Write a program to find out average of first n numbers using user defined function (UDF).	01
Write a program to declare structure student having member’s grade, name and roll number and access them in various ways.	01
Write a program using structure to get name, roll number, and marks of a student’s of a class and find out who got highest marks. Use concept of structure within structure.	01
Write a program to create an employee structure having member’s name, salary, Get data in employee structure through one function and display data using another function. Use concept of struct and function.	01
Write a program to declare and use pointer variables.	01
Write a program to swap two values with help of call by value and call by reference.	01
Write a program to find length of string using pointer and without using string functions.	01
Write a program to write the characters into file from standard input and then read the characters.	02
Write a program to write the integers into file from standard input and then read the integers.	02
Write a program that creates the structure of student and Scan the data of n students and store.	02

Write a program that copies the contents of one file into another.	02
Write a program that appends the content of file at the end of the other.	02

Reference Books:

1. Programming in ANSI C, Forth Edition, E Balagurusamy, TMH
2. Programming in C, Ashok Kamthane, Pearson
3. Let us C ,Y.P. Kanetkar, Infinity Science Press
4. C: The Complete Reference, Herbert Schildt, McGrawHill
5. Computer fundamentals and Programming in C, Pradip dey and Manas Ghosh, Oxford
6. Programming With C, by Byron Gottfried, Schaum's Outline Series, McGraw-Hill

Course Outcomes Mapping:

CO No.	Course Outcome (CO)	POs/ PSOs Mapped	Cognitive Level (RBT)	Knowledge Category	Lecture (Hrs)	Lab (Hrs)
CO1	Understand basic approaches and strategies used in computational problem solving.	PO1, PO2, PO12, PSO2	Understand	Conceptual	4	6
CO2	Recall and apply flowcharts and algorithms to represent solutions to simple problems.	PO1, PO2, PO5, PSO1	Remember, Apply	Factual, Procedural	5	8
CO3	Formulate logical solutions using flowcharts and algorithms for well-defined problems.	PO2, PO3, PO5, PSO1	Apply, Analyze	Procedural, Conceptual	5	10
CO4	Understand core concepts and terminology in the C programming language.	PO1, PO2, PO12, PSO2	Understand	Conceptual	4	8
CO5	Analyze program logic and apply control structures to implement modular and efficient code in C.	PO1, PO2, PO3, PO5, PSO1	Apply, Analyze	Conceptual, Procedural	6	14
CO6	Develop, test, and debug C programs using loops, functions, arrays, and decision-making constructs.	PO2, PO5, PO10, PO12, PSO1, PSO2	Apply, Create	Procedural, Metacognitive	6	14

Mapping of COs with POs & PSOs:

CO	PO												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	3	–	–	–	–	–	–	–	–	–	2	–	2
CO2	3	3	–	–	2	–	–	–	–	–	–	–	2	–
CO3	3	3	2	–	2	–	–	–	–	–	–	–	3	–
CO4	3	2	–	–	–	–	–	–	–	–	–	2	–	2
CO5	3	3	2	–	2	–	–	–	–	–	–	–	3	–
CO6	2	3	2	–	3	–	–	–	–	2	–	2	3	2

3: High, 2: Medium, 1: Low