

Computer Programming Lab
(Common to all Branches)

Course Code	23ES1152	Year	I	Semester	I
Course Category	Engineering Sciences	Branch	CSE (AIML)	Course Type	Lab
Credits	1.5	L-T-P	0-0-3	Prerequisites	Basic Mathematics
Continuous Internal Evaluation:	30	Semester End Exam:	70	Total Marks:	100

Course Outcomes		
Upon successful completion of the course, the student will be able to		
CO1	Apply C programming language constructs to solve the given problem	L2
CO2	Implement programs as an individual on different IDE's/ online platforms.	L3
CO3	Develop an effective report based on various programs implemented.	L3
CO4	Apply technical knowledge for a given problem and express it with effective oral communication.	L3
CO5	Analyze outputs using given constraints/test cases.	L4

Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (H:High, M: Medium, L:Low)														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3													
CO2	2				1									
CO3										3				
CO4										3				
CO5		3										1		

Syllabus		
Expt. No.	Contents	Mapped CO
I	<p>WEEK 1</p> <p>Objective: Getting familiar with the programming environment on the computer and writing the first program.</p> <p>Suggested Experiments/Activities:</p> <p>Tutorial 1: Problem-solving using Computers.</p> <p>Lab1: Familiarization with programming environment</p> <p>Basic Linux environment and its editors like Vi, Vim & Emacs etc.</p> <p>Exposure to Turbo C, gcc</p> <p>Writing simple programs using printf(), scanf()</p>	<p>CO1, CO2, CO3, CO4, CO5</p>
II	<p>WEEK 2</p> <p>Objective: Getting familiar with how to formally describe a solution to a problem in a series of finite steps both using textual notation and graphic notation.</p> <p>Suggested Experiments /Activities:</p> <p>Tutorial 2: Problem-solving using Algorithms and Flow charts. Lab 1:</p> <p>Converting algorithms/flow charts into C Source code.</p> <p>Developing the algorithms/flowcharts for the following sample programs</p> <p>Sum and average of 3 numbers</p> <p>Conversion of Fahrenheit to Celsius and vice versa</p> <p>Simple interest calculation</p>	<p>CO1, CO2, CO3, CO4, CO5</p>
III	<p>WEEK 3</p> <p>Objective: Learn how to define variables with the desired datatype, initialize them with appropriate values and how arithmetic operators can be used with variables and constants.</p> <p>Suggested Experiments/Activities:</p> <p>Tutorial 3: Variable types and type conversions:</p> <p>Lab 3: Simple computational problems using arithmetic expressions.</p> <p>Finding the square root of a given number</p> <p>Finding compound interest</p> <p>Area of a triangle using heron's formulae</p> <p>Distance travelled by an object</p>	<p>CO1, CO2, CO3, CO4, CO5</p>
IV	<p>WEEK 4</p> <p>Objective: Explore the full scope of expressions, type-compatibility of variables & constants and operators used in the expression and how operator precedence works.</p> <p>Suggested Experiments/Activities:</p> <p>Tutorial4: Operators and the precedence and as associativity:</p> <p>Lab4: Simple computational problems using the operator' precedence and associativity</p> <p>i) Evaluate the following expressions.</p> <p>a. $A+B*C+(D*E) + F*G$</p>	<p>CO1, CO2, CO3, CO4, CO5</p>

	<p>u. $A/B \ C-D+A \ D/S$</p> <p>c. $A+++B---A$</p> <p>d. $J = (i++) + (++i)$</p> <p>Find the maximum of three numbers using conditional operator</p> <p>Take marks of 5 subjects in integers, and find the total, average in float</p>	
V	<p>WEEK 5</p> <p>Objective: Explore the full scope of different variants of “if construct” namely if-else, null- else, if-else if*-else, switch and nested-if including in what scenario each one of them can be used and how to use them. Explore all relational and logical operators while writing conditionals for “if construct”.</p> <p>Suggested Experiments/Activities:</p> <p>Tutorial 5: Branching and logical expressions:</p> <p>Lab 5: Problems involving if-then-else structures.</p> <p>i) Write a C program to find the max and min of four numbers using if-else.</p> <p>ii) Write a C program to generate electricity bill.</p> <p>iii) Find the roots of the quadratic equation.</p> <p>iv) Write a C program to simulate a calculator using switch case.</p> <p>v) Write a C program to find the given year is a leap year or not.</p>	<p>CO1, CO2, CO3, CO4, CO5</p>
VI	<p>WEEK 6</p> <p>Objective: Explore the full scope of iterative constructs namely while loop, do-while loop and for loop in addition to structured jump constructs like break and continue including when each of these statements is more appropriate to use.</p> <p>Suggested Experiments/Activities:</p> <p>Tutorial 6: Loops, while and for loops</p> <p>Lab 6: Iterative problems e.g., the sum of series</p> <p>Find the factorial of given number using any loop.</p> <p>Find the given number is a prime or not.</p> <p>Compute sine and cos series</p> <p>Checking a number palindrome</p> <p>Construct a pyramid of numbers.</p>	<p>CO1, CO2, CO3, CO4, CO5</p>
VII	<p>WEEK 7:</p> <p>Objective: Explore the full scope of Arrays construct namely defining and initializing 1-D and 2-D and more generically n-D arrays and referencing individual array elements from the defined array. Using integer 1-D arrays, explore search solution linear search.</p> <p>Suggested Experiments/Activities:</p> <p>Tutorial 7: 1 D Arrays: searching.</p> <p>Lab 7: 1D Array manipulation, linear search</p> <p>Find the min and max of a 1-D integer array.</p> <p>Perform linear search on 1D array.</p> <p>The reverse of a 1D integer array</p> <p>Find 2's complement of the given binary number.</p> <p>Eliminate duplicate elements in an array.</p>	<p>CO1, CO2, CO3, CO4, CO5</p>

VIII	<p>WEEK 8 : Objective: Explore the difference between other arrays and character arrays that can be used as Strings by using null character and get comfortable with string by doing experiments that will reverse a string and concatenate two strings. Explore sorting solution bubble sort using integer arrays. Suggested Experiments/Activities: Tutorial 8: 2 D arrays, sorting and Strings. Lab 8: Matrix problems, String operations, Bubble sort i) Addition of two matrices ii) Multiplication two matrices iii) Sort array elements using bubble sort iv) Concatenate two strings without built-in functions v) Reverse a string using built-in and without built-in string functions</p>	CO1, CO2, CO3, CO4, CO5
IX	<p>WEEK 9: Objective: Explore pointers to manage a dynamic array of integers, including memory allocation & value initialization, resizing changing and reordering the contents of an array and memory de-allocation using malloc (), calloc (), realloc () and free () functions. Gain experience processing command-line arguments received by C Suggested Experiments/Activities: Tutorial 9: Pointers, structures and dynamic memory allocation Lab 9: Pointers and structures, memory dereference. i) Write a C program to find the sum of a 1D array using malloc() ii) Write a C program to find the total, average of n students using structures iii) Enter n students data using calloc() and display failed students list iv) Read student name and marks from the command line and display the student details along with the total. v) Write a C program to implement realloc()</p>	CO1, CO2, CO3, CO4, CO5
X	<p>WEEK 10: Objective: Experiment with C Structures, Unions, bit fields and self-referential structures (Singly linked lists) and nested structures Suggested Experiments/Activities: Tutorial 10: Bitfields, Self-Referential Structures, Linked lists Lab10 : Bitfields, linked lists Read and print a date using dd/mm/yyyy format using bit-fields and differentiate the same without using bit- fields i) Create and display a singly linked list using self-referential structure. ii) Demonstrate the differences between structures and unions using a C program. iii) Write a C program to shift/rotate using bitfields. Write a C program to copy one structure variable to another structure of the same type</p>	CO1, CO2, CO3, CO4, CO5
	<p>WEEK 11: Objective: Explore the Functions, sub-routines, scope and extent of variables, doing some experiments by parameter passing using call by value. Basic methods of numerical integration</p>	CO1, CO2, CO3, CO4, CO5

XI	<p>Suggested Experiments/Activities: Tutorial 11: Functions, call by value, scope and extent, Lab 11: Simple functions using call by value, solving differential equations using Eulers theorem.</p> <p>i) Write a C function to calculate NCR value. ii) Write a C function to find the length of a string. iii) Write a C function to transpose of a matrix. iv) Write a C function to demonstrate numerical integration of differential equations using Euler's method.</p>	
XII	<p>WEEK 12: Objective: Explore how recursive solutions can be programmed by writing recursive functions that can be invoked from the main by programming at-least five distinct problems that have naturally recursive solutions.</p> <p>Suggested Experiments/Activities: Tutorial 12: Recursion, the structure of recursive calls Lab 12: Recursive functions</p> <p>i) Write a recursive function to generate Fibonacci series. ii) Write a recursive function to find the lcm of two numbers. iii) Write a recursive function to find the factorial of a number. iv) Write a C Program to implement Ackermann function using recursion. v) Write a recursive function to find the sum of series.</p>	CO1, CO2, CO3, CO4, CO5
XIII	<p>WEEK 13: Objective: Explore the basic difference between normal and pointer variables, Arithmetic operations using pointers and passing variables to functions using pointers</p> <p>Suggested Experiments/Activities: Tutorial 13: Call by reference, dangling pointers Lab 13: Simple functions using Call by reference, Dangling pointers.</p> <p>i) Write a C program to swap two numbers using call by reference. ii) Demonstrate Dangling pointer problem using a C program. iii) Write a C program to copy one string into another using pointer. iv) Write a C program to find no of lowercase, uppercage, digits and other characters using pointers.</p>	CO1, CO2, CO3, CO4, CO5
XIV	<p>WEEK14: Objective: To understand data files and file handling with various file I/O functions. Explore the differences between text and binary files.</p> <p>Suggested Experiments/Activities: Tutorial 14: File handling Lab 14: File operations</p> <p>i) Write a C program to write and read text into a file. ii) Write a C program to write and read text into a binary file using fread() and fwrite()</p>	CO1, CO2, CO3, CO4, CO5
	<p>iii) Copy the contents of one file to another file. iv) Write a C program to merge two files into the third file using command-line arguments. v) Find no. of lines, words and characters in a file Write a C program to print last n characters of a given file.</p>	

Learning Resources**Text Books**

1. Ajay Mittal, Programming in C: A practical approach, Pearson.
2. Byron Gottfried, Schaum's Outline of Programming with C, McGraw Hill

Reference Books

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice-Hall of India
2. C Programming, A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE

e- Resources & other digital material

1. <https://www.geeksforgeeks.org/c-programming-language/>
2. <https://www.greatlearning.in/academy/learn-for-free/courses/c-programming>
3. https://onlinecourses.nptel.ac.in/noc22_cs101/course