

**Computer Programming Lab**  
(Common to all Branches)

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

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

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

<b>Syllabus</b>		
<b>Expt. No.</b>	<b>Contents</b>	<b>Mapped CO</b>
I	<p><b>WEEK 1</b></p> <p><b>Objective: Getting familiar with the programming environment on the computer and writing the first program.</b></p> <p>Suggested Experiments/Activities:</p> <p>Tutorial 1: Problem-solving using Computers.</p> <p>Lab1: Familiarization with programming environment</p> <p>i)Basic Linux environment and its editors like Vi, Vim &amp; Emacs etc.</p> <p>ii)Exposure to Turbo C, gcc</p> <p>iii)Writing simple programs using printf(), scanf()</p>	<p><b>CO1,</b> <b>CO2,</b> <b>CO3,</b> <b>CO4,</b> <b>CO5</b></p>
II	<p><b>WEEK 2</b></p> <p><b>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.</b></p> <p>Suggested Experiments /Activities:</p> <p>Tutorial 2: Problem-solving using Algorithms and Flow charts. Lab 1: Converting algorithms/flow charts into C Source code.</p> <p>Developing the algorithms/flowcharts for the following sample programs</p> <p>i)Sum and average of 3 numbers</p> <p>ii)Conversion of Fahrenheit to Celsius and vice versa</p> <p>iii)Simple interest calculation</p>	<p><b>CO1,</b> <b>CO2,</b> <b>CO3,</b> <b>CO4,</b> <b>CO5</b></p>
III	<p><b>WEEK 3</b></p> <p><b>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.</b></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>i)Finding the square root of a given number</p> <p>ii)Finding compound interest</p> <p>iii)Area of a triangle using heron's formulae</p> <p>iv)Distance travelled by an object</p>	<p><b>CO1,</b> <b>CO2,</b> <b>CO3,</b> <b>CO4,</b> <b>CO5</b></p>
IV	<p><b>WEEK 4</b></p> <p><b>Objective: Explore the full scope of expressions, type-compatibility of variables &amp; constants and operators used in the expression and how operator precedence works.</b></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. <math>A+B*C+(D*E) + F*G</math></p>	<p><b>CO1,</b> <b>CO2,</b> <b>CO3,</b> <b>CO4,</b> <b>CO5</b></p>

	<p>b. <math>A/B * C - B + A * D / 3</math>  c. <math>A+++B---A</math>  d. <math>J = (i++) + (++i)</math></p> <p>Find the maximum of three numbers using conditional operator  Take marks of 5 subjects in integers, and find the total, average in float</p>	
V	<p><b>WEEK 5</b>  <b>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”.</b>  Suggested Experiments/Activities:  Tutorial 5: Branching and logical expressions:  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.  ii) Write a C program to generate electricity bill.  iii) Find the roots of the quadratic equation.  iv) Write a C program to simulate a calculator using switch case.  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><b>WEEK 6</b>  <b>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.</b>  Suggested Experiments/Activities:  Tutorial 6: Loops, while and for loops  Lab 6: Iterative problems e.g., the sum of series</p> <p>i) Find the factorial of given number using any loop.  ii) Find the given number is a prime or not.  iii) Compute sine and cos series  iv) Checking a number palindrome  V) Construct a pyramid of numbers.</p>	<p>CO1,  CO2,  CO3,  CO4,  CO5</p>
VII	<p><b>WEEK 7:</b>  <b>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.</b>  Suggested Experiments/Activities:  Tutorial 7: 1 D Arrays: searching.  Lab 7: 1D Array manipulation, linear search</p> <p>i) Find the min and max of a 1-D integer array.  ii) Perform linear search on 1D array.  iii) The reverse of a 1D integer array  iv) Find 2's complement of the given binary number.  V) Eliminate duplicate elements in an array.</p>	<p>CO1,  CO2,  CO3,  CO4,  CO5</p>

VIII	<p><b>WEEK 8 :</b>  <b>Objective:</b> 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.  <b>Suggested Experiments/Activities:</b>  <b>Tutorial 8:</b> 2 D arrays, sorting and Strings.  <b>Lab 8:</b> 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><b>WEEK 9:</b>  <b>Objective:</b> Explore pointers to manage a dynamic array of integers, including memory allocation &amp; 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  <b>Suggested Experiments/Activities:</b>  <b>Tutorial 9:</b> Pointers, structures and dynamic memory allocation  <b>Lab 9:</b> 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><b>WEEK 10:</b>  <b>Objective:</b> Experiment with C Structures, Unions, bit fields and self-referential structures (Singly linked lists) and nested structures  <b>Suggested Experiments/Activities:</b>  <b>Tutorial 10:</b> Bitfields, Self-Referential Structures, Linked lists  <b>Lab10 :</b> 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><b>WEEK 11:</b>  <b>Objective:</b> 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><b>Suggested Experiments/Activities:</b>  <b>Tutorial 11:</b> Functions, call by value, scope and extent,  <b>Lab 11:</b> Simple functions using call by value, solving differential equations using Eulers theorem.  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><b>WEEK 12:</b>  <b>Objective:</b> 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.  <b>Suggested Experiments/Activities:</b>  <b>Tutorial 12: Recursion, the structure of recursive calls</b>  <b>Lab 12: Recursive functions</b>  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>	<p><b>CO1,</b>  <b>CO2,</b>  <b>CO3,</b>  <b>CO4,</b>  <b>CO5</b></p>
XIII	<p><b>WEEK 13:</b>  <b>Objective:</b> Explore the basic difference between normal and pointer variables, Arithmetic operations using pointers and passing variables to functions using pointers  <b>Suggested Experiments/Activities:</b>  <b>Tutorial 13: Call by reference, dangling pointers</b>  <b>Lab 13: Simple functions using Call by reference, Dangling pointers.</b>  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, uppercase, digits and other characters using pointers.</p>	<p><b>CO1,</b>  <b>CO2,</b>  <b>CO3,</b>  <b>CO4,</b>  <b>CO5</b></p>
XIV	<p><b>WEEK14:</b>  <b>Objective:</b> To understand data files and file handling with various file I/O functions. Explore the differences between text and binary files.  <b>Suggested Experiments/Activities:</b>  <b>Tutorial 14: File handling</b>  <b>Lab 14: File operations</b>  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()  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  vi)Write a C program to print last n characters of a given file.</p>	<p><b>CO1,</b>  <b>CO2,</b>  <b>CO3,</b>  <b>CO4,</b>  <b>CO5</b></p>

<b>Learning Resources</b>
<b>Text Books</b>
<ol style="list-style-type: none"><li>1. Ajay Mittal, Programming in C: A practical approach, Pearson.</li><li>2. Byron Gottfried, Schaum's Outline of Programming with C, McGraw Hill</li></ol>
<b>Reference Books</b>
<ol style="list-style-type: none"><li>1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice-Hall of India</li><li>2. C Programming, A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE</li></ol>
<b>e- Resources &amp; other digital material</b>
<ol style="list-style-type: none"><li>1. <a href="https://www.geeksforgeeks.org/c-programming-language/">https://www.geeksforgeeks.org/c-programming-language/</a></li><li>2. <a href="https://www.greatlearning.in/academy/learn-for-free/courses/c-programming">https://www.greatlearning.in/academy/learn-for-free/courses/c-programming</a></li><li>3. <a href="https://onlinecourses.nptel.ac.in/noc22_cs101/course">https://onlinecourses.nptel.ac.in/noc22_cs101/course</a></li></ol>