
COMPUTER PROGRAMMING LAB
AR23- B.Tech (Common to CIV, EEE, ME and ECE)
I-B.Tech., II-Semester

Course Objectives:

- To use basic data types, operators, expressions and expression evaluation mechanisms using C Programming Language.
- To implement control flows, construct in C Programming Language and understand the syntax, semantics and usability contexts of these different constructs.
- To develop composite data types in C and constructs available to develop their datatypes, utilize them to model things and dealing with data from and to external files.
- To design programs with different variations of the constructs available for practicing modular programming and understand the pros and cons of using different variants and apply optimization.

Course Outcomes:

Student should be able to:

CO1	Demonstrate the execution of programs written in C language	Apply
CO2	Choose the right control structure for solving the problem	Apply
CO3	Develop C programs which utilize the memory efficiently using programming constructs like pointers	Create
CO4	Develop, Debug and Execute programs to demonstrate the applications of arrays, functions, basic concepts of pointers in C	Create

UNIT-I

Exercise 1

Objective: Getting familiar with the programming environment on the computer and writing the first program.

Suggested Experiments/Activities:

Tutorial 1: Problem-solving using Computers.

Lab 1: Familiarization with programming environment

i) Basic Linux environment and its editors like Vi
ii) Exposure to gcc
iii) Getting familiar with Hackerrank platform
iv) Writing simple programs using printf(), scanf()

Exercise 2

Objective: Getting familiar with how to formally describe a solution to a problem in a series of finite steps using programming notation.

Suggested Experiments /Activities:

Tutorial 2: Problem-solving using C programming

Lab 2: Write a program using C language for the following sample problems

Part-A

i) Sum and average of 3 numbers
ii) Conversion of Fahrenheit to Celsius and vice versa

iii) Simple interest calculation

Part-B

iv) Debug the code to get desired output

Exercise 3

Objective: Learn how to define variables with the desired data-type, initialize them with appropriate values and how arithmetic operators can be used with variables and constants.

Suggested Experiments/Activities:

Tutorial 3: Variable types and type conversions:

Lab 3: Simple computational problems using arithmetic expressions.

i) Finding the square root of a given number
ii) Finding compound interest
iii) Distance travelled by an object
iv) Area of a triangle using heron's formulae

Exercise 4

Objective: Explore the full scope of expressions, type-compatibility of variables & constants and operators used in the expression and how operator precedence works.

Suggested Experiments/Activities:

Tutorial 4: Operators and the precedence, associativity:

Lab 4: Simple computational problems using the operator's precedence and associativity

i) Evaluate the following expressions.
    a. A+B*C+(D*E)+F*G
    b. A/B*C-B+A*D/3
    c. A+++B---A
    d. J= (i++)+(++i)
ii) Take marks of 5 subjects in integers, and find the total, average in float
iii) Find the maximum of three numbers using conditional operator

UNIT-II

Exercise 5

Objective: Explore the full scope of different variants of "if construct" namely if-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".

Suggested Experiments/Activities:

Tutorial 5: Branching and logical expressions:

Lab 5: Problems involving if-then-else structures.

Part-A

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) Write a C program to simulate a calculator using switch case.

Part-B

iv) Write a C program to find the given year is a leap year or not.
v) Find the roots of the quadratic equation.

Exercise 6
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.

Suggested Experiments/Activities:
Tutorial 6: Loops, while and for loops
Lab 6: Iterative problems e.g., the sum of series
Part-A
i) Find the factorial of given number using any loop.
ii) Find the given number is a prime or not.
iii) Checking a number palindrome
Part-B
iv) Construct a pyramid of numbers.

UNIT-III
Exercise 7:
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, 2 dimensional arrays and sorting solution bubble sort using integer arrays

Suggested Experiments/Activities:
Tutorial 7: 1D Arrays, 2D Arrays, Searching, Sorting.
Lab 7:1D Array manipulation, linear search
Part-A
i) Find the min and max of a 1-D integer array.
ii) Perform linear search on1D array.
iii) The reverse of a 1D integer array
iv) Addition of two matrices
Part-B
v) Sort array elements using bubble sort
vi) Multiplication two matrices

UNIT-IV
Exercise 8:
Objective: Explore the Functions, doing some experiments by parameter passing using call by value. Explore how recursive solutions can be programmed by writing recursive functions

Suggested Experiments/Activities:
Tutorial 8: Functions, call by value, Recursion, String operations
Lab 8: Simple functions using call by value, strings
Part-A
i) Write a C function to calculate nCr value
ii) Write a recursive function to find the factorial of a number
iii) Write a C function to find the length of a string
iv) Write a C function to reverse a string
Part-B
v) Write a user-defined C function to concatenate two strings
vi) Write a recursive function to generate Fibonacci series

vii) Write a recursive function to find the number of digits in a given non-zero number

Exercise 9:
Objective: Explore call by reference and passing variables to functions using pointers

Suggested Experiments/Activities:
Tutorial 9: Call by reference, dangling pointers
Lab 9: Simple functions using Call by reference, Dangling pointers
Part-A

i) Write a C program to swap two numbers using call by reference
ii) Write a C function to transpose of a matrix
iii) Write a C program to copy one string into another using pointer

Part-B

iv) Demonstrate Dangling pointer problem using a C program
v) Write a C program to find no of lowercase, uppercase, digits and other characters using pointers.

UNIT-V
Exercise 10:
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() functions.

Suggested Experiments/Activities:
Tutorial 10: Dynamic memory allocation
Lab 10: memory dereference.

i) Write a C program to find the sum of a 1D array using malloc()
ii) Write a program to read n students marks of a subject usng calloc() and find the number of students failed in the exam.
iii) Write a C program to implement realloc()

Exercise 11:
Objective: Experiment with C Structures, Unions, nested structures

Suggested Experiments/Activities:
Tutorial 11: Structures, Unions
Lab 11: Structures, Unions
Part-A

i) Write a C program to define a Employee(EmpId, Name, Salary, Experience) structure; read and display the details of two employees.
ii) Write a C program to find the total, average of n students using structures

Part-B

i) Demonstrate the differences between structures and unions using a C program.
ii) Write a C program to copy one structure variable to another structure of the same type.
iii) Read and print a date using dd/mm/yyyy format using structures

Exercise 12:
Objective: To understand data files and file handling with various file I/O functions. Explore the differences between text and binary files.

Suggested Experiments/Activities:
Tutorial 12: File handling
Lab 12: File operations
Part-A

i) Write a C program to write and read text into a file.
ii) Find no. of lines, words and characters in a file
iii) Copy the contents of one file to another file.
iv) Write a C program to print last n characters of a given file.

Part-B

v) Write a C program to write and read text into a binary file using fread() and fwrite()
vi) Write a C program to merge two files into the third file using command-line arguments.

Textbooks:

1. Programming for Problem Solving, Behrouz A. Forouzan, Richard F.Gilberg, Cengage.
2. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice-Hall of India

Reference Books:

1. Programming In C A-Practica1 Approach. Ajay Mittal, Pearson
2. Byron Gottfried, Schaum's Outline of Programming with C, McGrawHill

CO PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	3	1	1	-	1	1	-	1
CO2	3	2	1	1	3	1	1	-	1	1	-	1
CO3	2	2	2	3	3	1	1	-	1	1	-	1
CO4	2	2	2	3	3	1	1	-	1	1	-	1