Course learning outcomes

At the end of the course student will be able to:

1. Identify the parts of the computer system.
2. Adequately explain functioning of computer components.
3. Explain the process of problem solving using computer
4. Design an algorithmic solution for a given problem
5. Write a maintainable C program for a given algorithm.
6. Trace the given C program manually.
7. Write C program for simple applications of real life using structures and files.
8. Explain role of Operating system in computer system and applications of computer networks.

UNIT-I 12 Hours

Introduction to digital computers

Algorithms and Flowchart
Algorithms, flowcharts, writing algorithms and drawing flowcharts.

Overview of C language
Introduction, features, structure of C program, Compilation and Execution on Windows and Linux Platform.

Constants, Variables and Data types
Character set, C tokens, keywords and identifiers, constants, variables, data types, declaration of variables, Coding Standards.

Operators and Expressions
Arithmetic operators, logical operators, relational operators, assignment operators, Increment and decrement operators, conditional operators, bitwise operators, special operators, arithmetic expressions, evaluation of expressions, precedence of arithmetic operators, type conversion in expressions, operator precedence and associatively.

Managing Input / Output operations
Formatted and Unformatted input/output statements.

UNIT-II 12 Hours

Decision making and Branching
Decision making with if, if-else, nested if statements, else-if ladders, switch statement, ?: Operator, goto statement.

Looping
while statement, do while statement, for statement, jumps in loops.

Arrays
One dimensional arrays, declaration of one-dimensional arrays, initialization of one-dimensional arrays. Declaration of two-dimensional arrays, initialization of two-dimensional arrays.—Examples--

**Strings**
Introduction, Declaration of strings, initialization of strings, string-handling functions.

**UNIT-III**

12 Hours

**Introduction to pointers**
Definition, declaration, initialization of pointers, usage of pointers.

**User defined functions**
Need of user-defined functions, a multifunction program, elements of user defined functions, definition of function, return values and their types, function calls, function declaration, category of functions, no arguments and no return values, arguments but no return values, arguments with return values, no arguments but a return value, functions that return multiple values, nesting of functions, Introduction to recursion(factorial), passing arrays to functions, passing strings to functions, scope visibility and life time of variables, Command line arguments, multi-file programs.

**UNIT-IV**

12 Hours

**Introduction to structures**
Defining a structure, Declaring structure variables, accessing structure members, Initialization, Copying and comparing structure variables, Operations on individual members, array of structure .

**Introduction to files**
Defining and opening a file, closing a file, Input output operations on files.

**Introduction to parallel programming.**

**Hardware and Software**

**Operating system**
Definition, Purpose, types of operating systems, providing user interface, running programs, managing hardware, enhancing operating system with utility software

**Application / Networking**

**TEXT BOOKS:**
REFERENCE BOOKS:
Course learning outcomes

At the end of the course student will be able to:

1. Design an algorithmic solution for a given problem.
2. Draw flowcharts for the solution.
3. Write a maintainable C program for a given algorithm.
4. Write well documented and indented program according to coding standards..
5. Debug a given program.
6. Execute the C program.

TUTORIALS:

The students are made to execute the programs based on the similar concepts of assignments listed. The faculty will only guide the students in developing programs on their own. In each lab session based on the respective concept minimum two to three additional programs are to be discussed, developed and tested with every possible input set.

PART - I

Assignments for hands on experience

For Debugging: 05

For Indentation: 05

PART-II

1. Write a C program to find the largest of three numbers. (Nested if)
2. If cost and selling price of an item are the inputs, write a C program to determine whether the seller has made profit or incurred loss. Also determine how much profit he made or loss he incurred. (if-else)
3. Write a C program to find all the roots of a quadratic equation. (Using if – else and library functions)
4. Write a C program display the position of a given point with coordinates (x, y) on a plane (if-else and logical operators)
5. Write a C program to find area of circle, triangle, rectangle, square using switch statement
6. Write a C program to find and display the sum of first N natural numbers. Also compute and display the sum of odd and even numbers.(looping)
7. Write a C program to find the GCD and LCM of two integers numbers (use Euclid's Algorithm). (looping)
8. Write a C program to check whether the number is prime or not. Display appropriate message. (looping)
9. Write a program to print the multiplication table of the number entered by the user. The table should get displayed in the following form: (looping)
10. Write a C program to find the sum of the following series (looping):
\[
\sin(X) = X - \frac{X^3}{3!} + \frac{X^5}{5!} - \frac{X^7}{7!} + \ldots
\]

**PART- III**

1. Write a C program to read N integer numbers and arrange them in ascending order using bubble sort technique (Arrays, Sorting, Bubble sort).
2. Write a C program to sort a list of N integer numbers and search the given key element using binary search method. Display the result using the suitable message. (Searching, Binary search).
3. Write a C program to read a matrix of order M*N and find the sum of principal and secondary diagonal elements. (Matrix read print, primary and secondary diagonal elements of matrix.)
4. Write a C program to accept a string and reverse it without using library functions. Display the original and reversed string. (String handling)
5. Write a C program to read ‘N’ elements into an array and compute the sum of all the elements stored in an array using pointer. (Arrays and pointers)
6. Write a C program to find the factorial of a given integer number using recursive function. Accept number as command line argument. (Recursive function, command line arguments)
7. Write a C program to read list of integer numbers and find the mean, standard deviation and count number of integers less than mean of the list. Display all results in main function (Standard deviation, passing arrays to functions, returning one and more values from function). Use the following functions:
   i. To read given list of numbers.
   ii. To find mean and standard deviation (single function).
   iii. To find the number of elements those are less than the mean of that list.
8. Write a C program to read a matrix of order (M *N) and (P * Q) and compute the product of two matrices. (Passing matrix to functions). Use functions to:
   i. To read given matrices
   ii. To compute the product of two matrices
   iii. To print product matrix.
9. Write a C program to read n students information and display the information with Appropriate headings, where each student information consists of roll number, Name, total marks scored (Structure handling).
10. Write a C program to create a file of integers - intdata. Read intdata file and count Number of integers greater than 100 (Operations on files).

**NOTE:**

1. Each assignment in Part -I will be evaluated for 0.5 marks.
2. Each assignment in Part-II and PART- III will be evaluated for 1.25 marks.
3. The Continuous Internal Evaluation (CIE) is done for a total of 30 marks for assignments in Part A and B.
4. The lab test for CIE will be conducted for 20 marks and evaluated as per the following: Write-up: 5 marks, Execution of program: 10 marks, Viva: 5 marks.
5. In Semester End Exam (SEE), the student has to execute one assignment from each part and evaluated as per the following: Write up: 25% marks, Execution of program: 50% marks, Viva: 25% marks.