**PVP23**

# PRASAD V. POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

**(Autonomous)**

**KANURU, VIJAYAWADA-520007**

## I B. Tech – I Sem (Common to all)

## COMPUTER PROGRAMMING LAB

**SYLLABUS**

| | | | | | |
|---|---|---|---|---|---|
| **Course Code** | 23ES1152 | **Year** | I | **Semester** | I |
| **Course Category** | Engineering Sciences | **Branch** | CSE | **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 |

| | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PSO1** | **PSO2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | | | | | | | | | | | 1 | |
| **CO2** | | | | | 3 | | | | | | | 1 | |
| **CO3** | | | | | | | | | 1 | | | 1 | |
| **CO4** | 2 | | | | | | | | 2 | | | 1 | |
| **CO5** | | 2 | | | | | | | | | | 1 | |

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

| | | |
|---|---|---|
| | b. A/B*C-B+A*D/3<br>c. A+++B---A<br>d. J= (i++) + (++i)<br>ii) Find the maximum of three numbers using conditional operator<br>iii) Take marks of 5 subjects in integers, and find the total, average in float | |
| V | **WEEK 5**<br>**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".**<br>Suggested Experiments/Activities:<br>Tutorial 5: Branching and logical expressions:<br>Lab 5: Problems involving if-then-else structures.<br>i) Write a C program to find the max and min of four numbers using if-else.<br>ii) Write a C program to generate electricity bill.<br>iii) Find the roots of the quadratic equation.<br>iv) Write a C program to simulate a calculator using switch case.<br>v) Write a C program to find the given year is a leap year or not. | **CO1,CO2,CO3, CO4,CO5** |
| VI | **WEEK 6**<br>**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.**<br>Suggested Experiments/Activities:<br>Tutorial 6: Loops, while and for loops<br>Lab 6: Iterative problems e.g., the sum of series<br>i) Find the factorial of given number using any loop.<br>ii) Find the given number is a prime or not.<br>iii) Compute sine and cos series<br>iv) Checking a number palindrome<br>v) Construct a pyramid of numbers. | **CO1,CO2,CO3, CO4,CO5** |
| VII | **WEEK 7:**<br>**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.**<br>Suggested Experiments/Activities:<br>Tutorial 7: 1 D Arrays: searching.<br>Lab 7:1D Array manipulation, linear search<br>i) Find the min and max of a 1-D integer array.<br>ii) Perform linear search on1D array.<br>iii) The reverse of a 1D integer array<br>iv) Find 2's complement of the given binary number.<br>v) Eliminate duplicate elements in an array. | **CO1,CO2,CO3, CO4,CO5** |
| | **WEEK 8:** | **CO1,CO2,CO3,** |

| | | |
|---|---|---|
| VIII | **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.<br>**Suggested Experiments/Activities:**<br>**Tutorial 8:** 2 D arrays, sorting and Strings.<br>**Lab 8:** Matrix problems, String operations, Bubble sort<br>i) Addition of two matrices<br>ii) Multiplication two matrices<br>iii) Sort array elements using bubble sort<br>iv) Concatenate two strings without built-in functions<br>v) Reverse a string using built-in and without built-in string functions | **CO4,CO5** |
| IX | **WEEK 9:**<br>**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<br><br>**Suggested Experiments/Activities:**<br>**Tutorial 9:** Pointers, structures and dynamic memory allocation<br>**Lab 9:** Pointers and structures, memory dereference.<br>i) Write a C program to find the sum of a 1D array using malloc()<br>ii) Write a C program to find the total, average of n students using structures<br>iii) Enter n students data using calloc() and display failed students list<br>iv) Read student name and marks from the command line and display the student details along with the total.<br>v) Write a C program to implement realloc() | **CO1,CO2,CO3, CO4,CO5** |
| X | **WEEK 10:**<br>**Objective:** Experiment with C Structures, Unions, bit fields and self-referential structures (Singly linked lists) and nested structures<br>**Suggested Experiments/Activities:**<br>**Tutorial 10:** Bitfields, Self-Referential Structures, Linked lists<br>**Lab10 :** Bitfields, linked lists<br>Read and print a date using dd/mm/yyyy format using bit-fields and differentiate the same without using bit- fields<br>i) Create and display a singly linked list using self-referential structure.<br>ii) Demonstrate the differences between structures and unions using a C program.<br>iii) Write a C program to shift/rotate using bitfields.<br>Write a C program to copy one structure variable to another structure of the same type | **CO1,CO2,CO3, CO4,CO5** |
| | **WEEK 11:**<br>**Objective:** Explore the Functions, sub-routines, scope and extent of variables, | **CO1,CO2,CO3, CO4,CO5** |

| | | |
|---|---|---|
| XI | doing some experiments by parameter passing using call by value. Basic methods of numerical integration<br><br>**Suggested Experiments/Activities:**<br>**Tutorial 11:** Functions, call by value, scope and extent,<br>**Lab 11:** Simple functions using call by value, solving differential equations using Eulers theorem.<br>i) Write a C function to calculate NCR value.<br>ii) Write a C function to find the length of a string.<br>iii) Write a C function to transpose of a matrix.<br>iv) Write a C function to demonstrate numerical integration of differential equations using Euler's method. | |
| XII | **WEEK 12:**<br>**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.<br>**Suggested Experiments/Activities:**<br>**Tutorial 12: Recursion, the structure of recursive calls**<br>**Lab 12: Recursive functions**<br>i) Write a recursive function to generate Fibonacci series.<br>ii) Write a recursive function to find the lcm of two numbers.<br>iii) Write a recursive function to find the factorial of a number.<br>iv) Write a C Program to implement Ackermann function using recursion.<br>v) Write a recursive function to find the sum of series. | **CO1,CO2,CO3, CO4,CO5** |
| XIII | **WEEK 13:**<br>**Objective**: Explore the basic difference between normal and pointer variables, Arithmetic operations using pointers and passing variables to functions using pointers<br><br>**Suggested Experiments/Activities:**<br>**Tutorial 13: Call by reference, dangling pointers**<br>**Lab 13: Simple functions using Call by reference, Dangling pointers.**<br>i) Write a C program to swap two numbers using call by reference.<br>ii) Demonstrate Dangling pointer problem using a C program.<br>iii) Write a C program to copy one string into another using pointer.<br>iv) Write a C program to find no of lowercase, uppercase, digits and other characters using pointers. | **CO1,CO2,CO3, CO4,CO5** |
| XIV | **WEEK14:**<br>**Objective:** To understand data files and file handling with various file I/O functions. Explore the differences between text and binary files.<br>**Suggested Experiments/Activities:**<br>**Tutorial 14: File handling**<br>**Lab 14: File operations**<br>i) Write a C program to write and read text into a file.<br>ii) Write a C program to write and read text into a binary file using fread() and fwrite() | **CO1,CO2,CO3, CO4,CO5** |

| | | |
|---|---|---|
| | iii) Copy the contents of one file to another file.<br>iv) Write a C program to merge two files into the third file using command-line arguments.<br>v) Find no. of lines, words and characters in a file<br>vi) Write a C program to print last n characters of a given file. | |

| **Learning Resources** |
|---|
| **Text Books** |
| 1. Ajay Mittal, Programming in C: A practical approach, Pearson.<br>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<br>2. C Programming, A Problem-Solving Approach, Forouzan, Gilberg, Prasad, CENGAGE |
| **e- Resources & other digital material** |
| 1. https://www.geeksforgeeks.org/c-programming-language/<br>2. https://www.greatlearning.in/academy/learn-for-free/courses/c-programming<br>3. https://onlinecourses.nptel.ac.in/noc22_cs101/course |