#  **23A05101P-COMPUTER PROGRAMMING LAB**

## (Common to All branches of Engineering)

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Category:** | Engineering Science | **Credits:** | 1.5 |
| **Course Type:** | Practical | **Lecture-Tutorial-Practical:** | 0-0-3 |
| **Pre-requisite:** | BasicPrinciplesinComputers | **Sessional Evaluation:****External Exam Evaluation:****Total Marks:** | 3070100 |
| **Course****Objectives** | Thecourseaimstogivestudentshands–onexperienceandtrainthemontheconceptsofthe C-programminglanguage. |
| **Course****Outcomes** | Oncompletionofthiscourse,the students are able to :- |
| **CO1** | Read,understand,andtracetheexecutionofprogramswritteninClanguage. |
| **CO2** | Selectthe rightcontrolstructureforsolvingthe problem. |
| **CO3** | DevelopCprogramswhichutilizememoryefficientlyusingprogrammingconstructslikepointers. |
| **CO4** | Develop,DebugandExecuteprogramstodemonstratetheapplicationsofarrays,functions,basicconcepts ofpointersinC. |
| **Course Content** | **UNIT I****WEEK 1****Objective:** Getting familiar with the programming environment on the computer and writing the first program.**Suggested Experiments/Activities:****Tutorial1:** Problem-solving using Computers.**Lab1:**Familiarization with programming environment1. Basic Linux environment and its editors like Vi ,Vim &Emacsetc.
2. Exposure to Turbo C ,gcc
3. Writing simple programs using printf(),scanf()

**WEEK2****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.****Suggested Experiments/Activities:****Tutorial2:**Problem-solving using Algorithms and Flowcharts.**Lab1:**Converting algorithms/flowcharts into C Source code.Developing the algorithms/flowcharts for the following sample programs1. Sum and average of 3 numbers
2. Conversion of Fahrenheit to Celsius and vice versa
3. Simple interest calculation

**WEEK3****Objective:** Learn how to define variables with the desired data-type, initialize them with appropriate value sand how arithmetic operators can be used with variables and constants.**Suggested Experiments/Activities:****Tutorial3:** Variable types and type conversions:**Lab3:**Simple computational problems using arithmetic expressions.1. Finding the squareroot of a given number
2. Finding compound interest
3. Area of a triangle using heron’s formulae
4. Distance travelled by an object

**UNIT II****WEEK 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:****Tutorial4:** Operators and the precedence and as associativity:**Lab4:** Simple computational problems using the operator’s precedence and associativity1. Evaluate the following expressions.
	1. A+B\*C+(D\*E)+F\*G
	2. A/B\*C-B+A\*D/3

c.A+++B---Ad.J=(i++)+(++i)1. Find the maximum of three numbers using conditional operator
2. Take marks of 5 subjects in integers, and find the total, average in float

**WEEK5****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”.**Suggested Experiments/Activities:****Tutorial5:**Branching and logical expressions:**Lab5**:Problems involving if-then-else structures.1. Write a C program to find the max and min off our numbers using if-else.
2. Write a C program to generate electricity bill.
3. Find the roots of the quadratic equation.
4. Write a C program to simulate a calculator usings witch case.
5. Write a C program to find the given year is a leap year or not.

**WEEK6****Objective:** Explore the full scope of iterative constructs namely whileloop, do-whilelo opand for loop in addition to structured jump constructs like break and continue including when each of these statements is more appropriate touse.**Suggested Experiments/Activities:****Tutorial6:**Loops,whileandfor loops**Lab 6:**Iterative problems e.g.,the sum of series1. Find the factorial of given number using any loop.
2. Find the given number is a prime or not.
3. Compute sine and cos series
4. Checking a number palindrome
5. Construct a pyramid of numbers.

**UNIT III****WEEK 7****Objective:** Explore the full scope of Arrays construct namely defining and initializing 1-Dand 2-D and more generically n-D arrays and referencing individual array elements from thedefined array.Usinginteger1-Darrays,explore search solution linear search.**Suggested Experiments/Activities:****Tutorial7:**1DArrays:searching.**Lab7:**1DArray manipulation,linear search1. Find the min and max of a 1-D integer array.
2. Perform linear search on 1 D array.
3. The reverse of a 1D integer array
4. Find 2’scomplement of the given binary number.
5. Eliminate duplicate elements in an array.

**WEEK8:****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:****Tutorial8:**2Darrays, sorting and Strings.**Lab8:**Matrix problems,String operations,Bubble sort1. Addition of two matrices
2. Multiplication tw omatrices
3. Sort array elements using bubble sort
4. Concatenate two strings without built-in functions
5. Reverse a string using built-in and without built-in string functions

**UNIT IV****WEEK 9****Objective:** 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**Suggested Experiments/Activities:****Tutorial9:** Pointers, structures and dynamic memory allocation**Lab9:**Pointers and structures ,memory dereference.1. Write a C program to find the sum ofa 1D array using malloc()
2. Write a C program to find the total, average of n students using structures
3. Enter n students data using calloc() and display failed students list
4. Read student name and marks from the command line and display the student details along with the total.
5. Write a C program to implement realloc()

**WEEK10:****Objective:** Experiment with C Structures, Unions ,bit fields and self-referential structures (Singly linked lists) and nested structures**Suggested Experiments/Activities:****Tutorial10:** Bit fields, Self-Referential Structures, Linked lists**Lab10:**Bitfields,linkedlistsRead and print a date using dd/mm/yyyy format using bit-fields and differentiate the same without using bit-fields1. Create and display a singly linked list using self-referential structure.
2. Demonstrate the differences between structures and unions using a C program.
3. Write a C program to shift/rotate using bit fields.
4. Write a C program to copy one structure variable to another structure of the same type.

**UNIT V****WEEK 11****Objective:** Explore the Functions ,sub-routines ,scope and extent of variables, doing some experiments by parameter passing using callby value. Basic methods of numerical integration**Suggested Experiments/Activities:****Tutorial11:**Functions, callby value, scope and extent,**Lab11:** Simple functions using callby value, solving differential equations using Eulers theorem.1. Write a C function to calculate NCR value.
2. Write a C function to find the length of a string.
3. Write a C function to transpose of a matrix.
4. Write a C function to demonstrate numerical integration of differential equations using Euler’s method

**WEEK12:****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.**Suggested Experiments/Activities:****Tutorial12:**Recursion, the structure of recursive calls**Lab12:**Recursive functions1. Write a recursive function to generate Fibonacci series.
2. Write a recursive function to find the lcm of two numbers.
3. Write a recursive function to find the factorial of a number.
4. Write a C Program to implement Ackermann function using recursion.
5. Write a recursive function to find the sum of series.

**WEEK13:****Objective:** Explore the basic difference between normal and pointer variables, Arithmetic operations using pointers and passing variables to functions using pointers**Suggested Experiments/Activities:****Tutorial13:**Callbyreference,danglingpointers**Lab13:** Simple functions using Callby reference, Dangling pointers.1. Write a C program to swap two numbers using callby reference.
2. Demonstrate Dangling pointer problem using a C program.
3. Write a C program to copy one string in to another using pointer.
4. Write a C program to find no of lowercase, uppercase, digits and other characters using pointers.

**WEEK14:****Objective:** To understand data files and file handling with various file I/O functions. Explore the differences between extand binary files.**Suggested Experiments/Activities:****Tutorial14:**Filehandling**Lab14:** File operations1. Write a C program to write and read text into a file.
2. Write a C program to write and read text into a binary file using fread() and fwrite()
3. Copy the contents of one file to another file.
4. Write a C program to merge two files in to the third file using command-line arguments.
5. Find no.of lines, words and characters in a file
6. vi) Write a C program to print last n characters of a given file.

 |
| **Text Books** | 1. AjayMittal,ProgramminginC:Apractical approach, Pearson.
2. ByronGottfried,Schaum&#39;sOutlineofProgramming withC,McGrawHill.
 |
| **Reference Books** | 1. BrianW.KernighanandDennisM.Ritchie,TheCProgrammingLanguage,Prentice-HallofIndia
2. CProgramming,AProblem-SolvingApproach,Forouzan,Gilberg,Prasad,CENGAGE
 |

|  |
| --- |
| Contribution of Course Outcomes towards achievement of Program Outcomes (3-High, 2-Medium, 1-Low) |
|  | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 |
| CO1 | 3 | 3 | 2 | - | - | - | - | 2 | 2 | 3 | - | - | 3 | 2 |
| CO2 | 3 | 3 | 2 | - | - | - | - | 2 | 2 | 2 | - | - | 3 | 2 |
| CO3 | 3 | 3 | 2 | - | - | - | - | 2 | 2 | 3 | - | - | 3 | 2 |
| CO4 | 3 | 3 | 2 | - | - | - | - | 2 | 2 | 3 | - | - | 3 | 2 |