COURSE TITLE   :   **PROGRAMMING IN C LAB**
COURSE CODE   :   **2139**
COURSE CATEGORY   :   **Basic Engineering**
PERIODS/WEEK   :   **3**
PERIODS/SEMESTER   :   **45**
CREDITS   :   **2**

Hardware Requirement : Desk Top Computer.

Software Requirement : : Linux Operating System with GCC

Course General Outcomes:

At the end of the Course, the students will be able to
- Develop the logic to solve the given problem.
- Understand the concepts of constants, variables, data types and operators.
- Develop programs using input and output operations.
- Write programs using decision control structures in C.
- Write programs using loop control structures in C.
- Write programs using case control structures in C.
- Write programs based on arrays.
- Write Programs using string handling functions.
- Write programs using user-defined functions,
- Write programs using structures
- Write programs using the concept of Pointers.

List of Experiments

1. **To apply programming concepts in C**
   a. Demonstrate output functions and input function for a simple application.
   b. *Write programs for simple expression evaluation.*

Create a Sample application for inputting details such as regno, branch code, semester and 5 marks of a student, calculate and display total mark along with student details.
   c. Write programs using simple if statement.
   d. Write programs using if..else, elseif and nested if.
   e. Write programs using switch statement.

*Sample Experiment :: Modify the sample application to calculate and display the **grade** based on the total mark along with student details*.

2. **To apply Looping controls and arrays**
   a. Write programs using while, do..while and for statements.

   Sample Experiment :: Modify the sample application to calculate and display the details along with the total mark and grade of 'n' number of students.

   b. Write programs using one dimensional array.

c. Write programs to perform one dimensional operations like

Insert, delete, search, sort, largest, smallest, second largest, and compute sum and

average of array elements

d. . Write programs using two dimensional arrays.

e. Write programs to perform two dimensional array operations like

Transpose of a matrix, checking of symmetric matrix, sum of elements of matrix, row sum,

column sum, sum of diagonal elements, matrix addition, matrix multiplication

Sample Experiment :: Modify the sample application to store the details of 'n' students into multiple arrays, calculate and display the total mark along with the student details in the order of total mark.

3. To apply Pointers and Strings

1.1 Write programs to apply pointers
1.2 Write programs for string manipulations

Sample Experiment :: Modify the sample application to store the details of 'n' students with following data elements regno, name (character array), branch, semester and 5 marks (integer array) and display student details along with total mark and grade based on boundary conditions.

4.To apply Functions

a. Write simple programs based on library functions
b. Write programs to illustrate 'User-defined functions"
c. Write Function subprograms using arrays and pointers
d. Write Function subprograms to illustrate array as argument
e. Write Function subprograms to Illustrate pointers as arguments

Sample Experiment :: Modify the sample application to store the details of n number of students in multiple arrays with following data elements – regno, name, branch, semester and 5 marks and display the student details along with total mark and grade based on boundary condition. Use function for grade calculation. Display the output with proper headings and format.

f. Write programs to illustrate structure

Sample Experiment :: Modify the sample application by creating a structure named student_details and display above mentioned students details.