# CL-103
# Computer Programming
# Lab # 11 Sec D

**Objectives:**

- Friend function
- Friend Classes
- Abstract Classes

**Note: Carefully read the following instructions (*Each instruction contains weightage*)**

1. There must be a block of comments at start of every question's code; the block should contain brief description about functionality of code
2. Proper indentation of code is essential
3. Variable name should be meaningful
4. Make a Microsoft Word file and past all of your C++ code with screenshot of outputs in MS word.
5. First think about statement problems and then write/draw your logic on copy.
6. After copy pencil work, code the problem statement on MS Studio C++ compiler.
7. At the end when you done your today lab tasks, attached only MS word file and make your submission on slate.
8. Late and email submission is not accepted. All tasks must be submitted during the lab time.

**Problem 1: (Polymorphism, abstract and concrete classes)**

Design a class Employee with subclasses Piece Worker and Hourly Worker. A Piece Worker represents an employee whose pay is based on then number of pieces of merchandise produced. An Hourly Worker represents an employee whose pay is based on an hourly wage and then number of hours worked. Hourly workers receive overtime pay (1.5times the hourly wage) for all hours worked in excess of 40 hours. Class Piece Worker should contain private instance variables wage (to store the employee's wage per piece) and pieces (to store the number of pieces produced). Class Hourly Worker should contain private instance variables wage (to store the employee's wage per hour) and hours (to store the hours worked). In class Piece Worker, provide a concrete implementation of method earnings that calculates the employee's earnings by multiplying the number of pieces produced by the wage per piece. In class Hourly Worker, provide a concrete implementation of method earnings that calculates the employee's earnings by multiplying the number of hours worked by the wage per hour. If the number of hours worked is over40, be sure to pay the Hourly Worker for the overtime hours. Add a pointer to an object of each new class into the vector of Employee pointers in main. For each Employee, display its string representation and earnings.

**Problem 2: (Polymorphism, Pure Virtual function, abstract and concrete classes)**

Finding the areas of Polygon family (Square, Rectangle and Triangle) using Polymorphism and Pure Virtual function.
Define a base class Polygon having the following attributes:

1. Float length
2. Float width

Pure Virtual Member functions:

1. Float Area() to calculate Area
2. Float Perimeter() to calculate Perimeter
3. Void Display() to display the calculated Area and Perimeter

Inherit the following classes from Polygon publically (Inherit all data members and member functions):

1. Square
2. Rectangle
3. Triangle

**Task:**

1. Write default constructor and Overloaded constructor to overload the values of Length and Width.
2. Calculate the Areas and Perimeters of Polygon family.

3. Show the results on the console using Display () function.

4. Write base class pointer "*Bptr" which points to the objects of derived classes one by one and display the area and perimeter of subclasses using Dynamic Binding

**Hints:**

- Square Area/Perimeter=4*L
- Rectangle A=2*(LxW) P=2*(L+W)
- Triangle A= (LxW)/2 P= (L+W+H)

## Problem 3: (friend function)

Use friend function to write a program for employee salary of Sitara Private Limited.

Make a class and define following data members:

- Name of Employee
- Rank of employee (Basic pay scale)
- Basic pay
- MD (Medical allowance)
- HR (House Rent)
- Gross pay (total pay)

**Task 1:** Define a private Member function of class

- Display ( );

Display complete record of employee including Name of employee, rank, basic pay and gross pay and Next year pay using this pointer.

**Task 2:** Also define following functions

- Read_record ( ); (friend function)

To take record of employee like Name of employee, Rank, basic pay from user.

- Gross_pay ( ); (friend function)

Calculate the gross pay of employee using basic pay and allowances.

- Annual_increment ( ); (Friend function)

In the next year the pay of employee increases 20% of basic pay.

**Task 3:** Use these friend functions by accessing private function Display of your class

Note:

- Medical allowance value is 60% of basic pay and House rent have 28.9% of basic pay.
- Write a constructor in which data members should be initialized and display message "Constructor".
- Use efficient data type (string, int, float etc.)
- At the end define destructor and display message "Destructor".

**Problem 4: (friend Classes)**

**Task1:**
Write a class Base having the following integer attributes as:

1. Base1
2. Base2
3. Base3 (private)

Write a default constructor for Base that should initialize all the attributes to 0. Write an overloaded constructor that should initialize all the attributes to the passed integer. Write a display method to display the attributes of Base class.

**Task2:**
Publicly inherit a class Derived from Base. Derived should have an integer attribute Derived1 and Derived2 (private). Write default constructor for Derived that should initialize all the attributes (inherited and self) to 1. Write an overloaded constructor that should initialize all the attributes to the passed integer.

**Task3:**
Write a class Friend_Class. Make it a friend of Base class.
Write a method AccessMethod in Friend_Class class.
Access the members of Base class in this method (public as well as private).

**Task4:**
Write a class Friend1_Class. Make it a friend of Derived class.
Write a method AccessMethod1 in Friend1_Class class.
Access the members of Derived class in this method.

**Task5:**
Write a class Friend2_Class. Make it a friend of Friend_Class.
Write a method AccessMethod in this Friend2_Class class.
Access the members of Friend_Class in this method and also the members of Base class (check whether you can access its public or private members)