Programming Paradigms Lab Assignment (CS453)

Assignment Sheet 4 : Inheritance and Polymorphism using C++

Time : Two weeks

Develop the below mentioned programs using these concepts wherever applicable -

- Any C++ concepts as mentioned in Assignment Sheet 4
- Inheritance
- Polymorphism

Problems

1. Complete the classes below with suitable data members and methods. In client program take instance of these classes and demonstrate various functionality of these objects.

```
Class Vehicle {
     private :
          //  Price
          //  Manufacturer
     public :
          //  Default constructor
          //  Parametrized constructor
          //  Copy constructor
          //  Assignment operator
          //  Read data
          //  Display data
};
```

```
Class Car : public Vehicle {
     private :
          //  Color
          //  NumberOfSeats
          //  Model
     public :
          //  Default constructor
```

```
            //  Parametrized constructor
            //  Copy constructor
            //  Assignment operator
            //  Read data
            //  Display data
}
```

2. Write a program to design Classes for `Student, Clerk, Professor`. Each of these Classes should contain below mentioned attributes. Make sure proper class hierarchy is designed following the principle of inheritance.

   Provide a mechanism to display the profile/detail of various kind of Object of these class.

   `Student` : Name, Age, Gender, Dept, Year

   `Clerk` : Name, Age, Gender, WorkLoad, Salary

   `Professor` : Name, Age, Gender, Dept, CourseLoad, Salary

3. A plot is broken into different geometric shapes like `Triangle`, `Rectangle` and `Circle` of different size of arbitrary number. Provide a mechanism to sum up total area covered by these shapes.

Once problem 1, 2 and 3 are completed, then attempt problem 4 and 5.

4. Reuse the code from program # 3 above and implement below program -

   Design a class `ShapeStack` that can store various kinds of geometric shapes like `Triangle`, `Rectangle` and `Circle`. Make sure `Push(...)` and `Pop(...)` operations are not overloaded (such as single interface per operation). Also provide `Display(...)` and `TotalAreaCovered(...)` method under `ShapeStack` class.

5. Create a class `Cricketer` with required data members. Inherit the class and create two derived classes `Batsman` and `Bowler` with required data members and member functions. Create a class `Allrounder` which is derived from both `Batsman` and `Bowler` classes.

   Implement the following functions in `Allrounder` class:
   1. Insert records match wise records
   2. Count batting average
   3. Count total wickets
   4. Find highest wickets score against which country
   5. Find highest runs score against which country