Lab 8 Classes -II

Problem 1:

Design a class Complex for handling Complex numbers and include the following **private** data members:

- real: a double
- imaginary: a double

The class has the following member functions.

1. A constructor initializing the number with **default parameters**.
2. Overloaded Constructors.
   - Complex(double r, double i)
     Note*:Use member function initialization for all data members.
   - Complex(Complex & copy) // copy constructor

3. Getters and Setters of the class data members as given below
   - void setReal(double r)
   - double getReal()
   - void setImaginary(double i)
   - double getImaginary()

4. Overload the following member function in the class

   - **Complex addComplex(double r)**
     It adds r of type double to real part of complex number while imaginary part remains same. And returns newly generated complex number.

   - **Complex addComplex(Complex &c1)**
     It adds both complex numbers and returns newly generated complex number.

   - **Complex subComplex(double r)**
     It subtracts r of type double from real part of complex number while imaginary part remains same. And returns newly generated complex number.

   - **Complex subComplex(Complex &c1)**
     It subtracts both complex numbers and returns newly generated complex number.

   - **Complex mulComplex(double n)**
     It's a scalar multiplication. Real and imaginary parts are multiplied by n. and returns newly generated complex number.

   - **Complex mulComplex(Complex &c1)**
     It multiplies both complex numbers and returns newly generated complex number. (a+bi)(c+di) = (ac−bd) + (ad+bc)i

Problem 2:

Design a class **Holiday** that represents a holiday during the year. This class has three **private** data members:

- **name**: A string that represents the name of holiday.
- **day**: An integer that holds the day of the month of holiday.
- **month**: A string that holds the month the holiday is in.

1. Write a default constructor that initializes each data member of class such that **name** with **NULL**, **day** with **0** and **month** with **NULL**
   Holiday()

2. Write a constructor that accepts the arguments for each data member such that **string n** assigned to **name**, **int d** to **day** and **string m** to **month**.
   Holiday(string &n, int d, string &m)
   Note*:Use member function initialization for all data members.

3. Generate getter setter of each member variable: such that **name** should never be greater than 50 characters, **day** should never be negative and **month** should not be greater than 10 characters.
   - bool setName(string &s)
   - string getName()
   - bool setDay(int u)
   - int getDay()
   - bool setMonth(string &p)
   - string getMonth()

4. Write a function **inSameMonth** (outside class) which takes two Holiday objects as arguments, compares two objects of the class Holiday, and returns true if they have the same month otherwise false.
   bool inSameMonth (Holiday &a, Holiday &b)

5. Write a function **avgDate** (outside class) which takes an **array** of type Holiday and its **size** as its argument and returns a **double** value that is the average of the entire day data member in the Holiday array **arr**. You may assume that the array is full (i.e. does not have any NULL entries).
   double avgDate(Holiday arr[], int size)

Problem 3:

Write a class **Point** with data members

- **x** : a integer x coordinate
- **y** : a integer x coordinate

The class has the following member functions.

1. Default constructor initializing the coordinates to zero
   Point()

2. A constructor that takes the values and initializes coordinates with x1 and y1
   Point (int x1, int y1)

   Note*:Use member function initialization for all data members.

3. A constructor that takes Point's reference and initializes coordinates
   Point (Point &copy) // it's a copy constructor

4. a destructor that prints the following statement on screen :"Destructor Called" Point()
5. Getter and Setter functions for **x** and **y**.