Assignment 5

PersonData and CustomerData Classes	50 pts.
Shapes	50 pts.
TOTAL	**100 pts.**

Part 1 (Class 9 – Inheritance)

PersonData and CustomerData Classes

PersonData Class

Design a class named PersonData with the following member variables:

- `lastName`
- `firstName`
- `address`
- `phone`

All member variables are of type string.

- Write appropriate accessor and mutator functions for these member variables.
- Write a default constructor that sets all variables to empty strings.
- Write a non-default constructor that takes all the data about customer as parameters (4 parameters) and initializes member variables.

CustomerData Class

Design a class named CustomerData, which is derived from the PersonData class. The CustomerData class should have the following member variables:

- `customerNumber`
- `mailingList`

The customerNumber variable will be used to hold a unique integer for each customer. The mailingList variable should be a bool. It will be set to true if the customer whishes to be on a mailing list, or false if the customer does not with to be on a mailing list.

- Write appropriate accessor and mutator functions for these member variables.
- Write a default constructor that sets all variables (including inherited ones) to empty strings.
- Write a non-default constructor that takes all the data about customer as parameters (name, address, etc., total of 6 parameters) and initializes member variables.

Demonstrate an object of CustomerData class in a simple program.

Part 2 (Class 10 - Polymorphism and Virtual Functions)

Shapes

Define a pure abstract base class called `BasicShape`. The `BasicShape` class should have the following members:
Protected member variable:
`area`, a double used to hold the shape's area.
Public member functions:
`getArea` , accessor function for the area.
`calcArea`, pure virtual function.

Next, define a class named `Circle`. It should be derived from the `BasicShape` class. It should have the following members:
Private member variables:
`centerX`, integer to hold x coordinate of the circle's center.
`centerY`, integer to hold y coordinate of the circle's center.
`radius`, integer to hold the radius of the circle.
Public member functions:
Default constructor
constructor – accepts values for centerX, centerY, and radius. Should call the overridden `calcArea` function described below.
`getCenterX` – returns the value in centerX.
`getCenterY` – returns the value in centerY.
`calcArea` – calculates the area of the circle (area = 3.14*radius*radius) and stores it in the `area` member variable.

Next, define a class named `Rectangle`. It should be derived from the `BasicShape` class. It should have the following members:
Private member variables:
`width`, integer to hold the width of the rectangle.
`length`, integer to hold the length of the rectangle.
Public member functions:
Default constructor
constructor – accepts values for length and width. Should call the overridden `calcArea` function described below.
`getLength` – returns the value in length.
`getWidth` – returns the value in width.

`calcArea` – calculates the area of the rectangle (area = length*width) and stores it in the `area` member variable.

After you have created these classes, create a driver program that defines a `Circle` object and a `Rectangle` object. Demonstrate that each object properly calculates and reports its area.