

## Cloud and Web Services Lab

### Assignment -1

#### **Ques 1. Employee and ProductionWorker Classes**

Design a class named Employee. The class should keep the following information:

- Employee name
- Employee number
- Hire date

Write one or more constructors, and the appropriate accessor and mutator functions, for the class. Next, write a class named ProductionWorker that is derived from the Employee class. The ProductionWorker class should have member variables to hold the following information:

- Shift (an integer)
- Hourly pay rate (a double)

The workday is divided into two shifts: day and night. The shift variable will hold an integer value representing the shift that the employee works. The day shift is shift 1, and the night shift is shift 2. Write one or more constructors, and the appropriate accessor and mutator functions, for the class. Demonstrate the classes by writing a program that uses a ProductionWorker object.

#### **ShiftSupervisor Class**

In a particular factory, a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals. Design a ShiftSupervisor class that is derived from the Employee class. The ShiftSupervisor class should have a member variable that holds the annual salary, and a member variable that holds the annual production bonus that a shift supervisor has earned. Write one or more constructors and the appropriate accessor and mutator functions for the class. Demonstrate the class by writing a program that uses a ShiftSupervisor object.

In a particular factory, a team leader is an hourly paid production worker who leads a small team. In addition to hourly pay, team leaders earn a fixed monthly bonus. Team leaders are required to attend a minimum number of hours of training per year. Design a TeamLeader class that extends the ProductionWorker class. The TeamLeader class should have member variables for the monthly bonus amount, the required number of training hours, and the number of training hours that the team leader has attended. Write one or more constructors and the appropriate accessor and mutator functions for the class. Demonstrate the class by writing a program that uses a TeamLeader object.

#### **Ques 2. PersonData and CustomerData Classes**

Design a class named PersonData with the following member variables:

- lastName
- firstName
- address
- city
- state
- zip
- phone

Write the appropriate accessor and mutator functions for these member variables. Next, 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 wishes to be on a mailing list, or `false` if the customer does not wish to be on a mailing list. Write appropriate accessor and mutator functions for these member variables.

Demonstrate an object of the `CustomerData` class in a simple program.

### **PreferredCustomer Class**

A retail store has a preferred customer plan where customers may earn discounts on all their purchases. The amount of a customer's discount is determined by the amount of the customer's cumulative purchases in the store.

- When a preferred customer spends \$500, he or she gets a 5 percent discount on all future purchases.
- When a preferred customer spends \$1,000, he or she gets a 6 percent discount on all future purchases.
- When a preferred customer spends \$1,500, he or she gets a 7 percent discount on all future purchases.
- When a preferred customer spends \$2,000 or more, he or she gets a 10 percent discount on all future purchases.

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

- `purchasesAmount` (a `double`)
- `discountLevel` (a `double`)

The `purchasesAmount` variable holds the total of a customer's purchases to date. The `discountLevel` variable should be set to the correct discount percentage, according to the store's preferred customer plan. Write appropriate member functions for this class and demonstrate it in a simple program.

*Input Validation: Do not accept negative values for any sales figures.*

### **Ques 3. Ship, CruiseShip, and CargoShip Classes**

Design a `Ship` class that has the following members:

- A member variable for the name of the ship (a `string`)
- A member variable for the year that the ship was built (a `string`)
- A constructor and appropriate accessors and mutators
- A virtual `print` function that displays the ship's name and the year it was built.

Design a `CruiseShip` class that is derived from the `Ship` class. The `CruiseShip` class should have the following members:

- A member variable for the maximum number of passengers (an `int`)
- A constructor and appropriate accessors and mutators
- A `print` function that overrides the `print` function in the base class. The `CruiseShip` class's `print` function should display only the ship's name and the maximum number of passengers.

Design a `CargoShip` class that is derived from the `Ship` class. The `CargoShip` class should have the following members:

- A member variable for the cargo capacity in tonnage (an `int`)
- A constructor and appropriate accessors and mutators

- A print function that overrides the print function in the base class. The CargoShip class's print function should display only the ship's name and the ship's cargo capacity. Demonstrate the classes in a program that has an array of Ship pointers. The array elements should be initialized with the addresses of dynamically allocated Ship, CruiseShip, and CargoShip objects.

**Ques 4.** Define a pure abstract base class called BasicShape. The BasicShape class should have the following members:

Private Member Variable:

area: A double used to hold the shape's area.

Public Member Functions:

getArea: This function should return the value in the member variable area.

calcArea: This function should be a 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: a long integer used to hold the x coordinate of the circle's center

centerY: a long integer used to hold the y coordinate of the circle's center

radius: a double used to hold the circle's radius

Public Member Functions:

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 ( $\text{area} = 3.14159 * \text{radius} * \text{radius}$ ) and stores the result in the inherited member area.

Next, define a class named Rectangle. It should be derived from the BasicShape class. It should have the following members:

Private Member Variables:

width: a long integer used to hold the width of the rectangle

length: a long integer used to hold the length of the rectangle

Public Member Functions:

constructor: accepts values for width and length. Should call the overridden calcArea function described below.

getWidth: returns the value in width.

getLength: returns the value in length.

calcArea: calculates the area of the rectangle ( $\text{area} = \text{length} * \text{width}$ ) and stores the result in the inherited member area. 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.