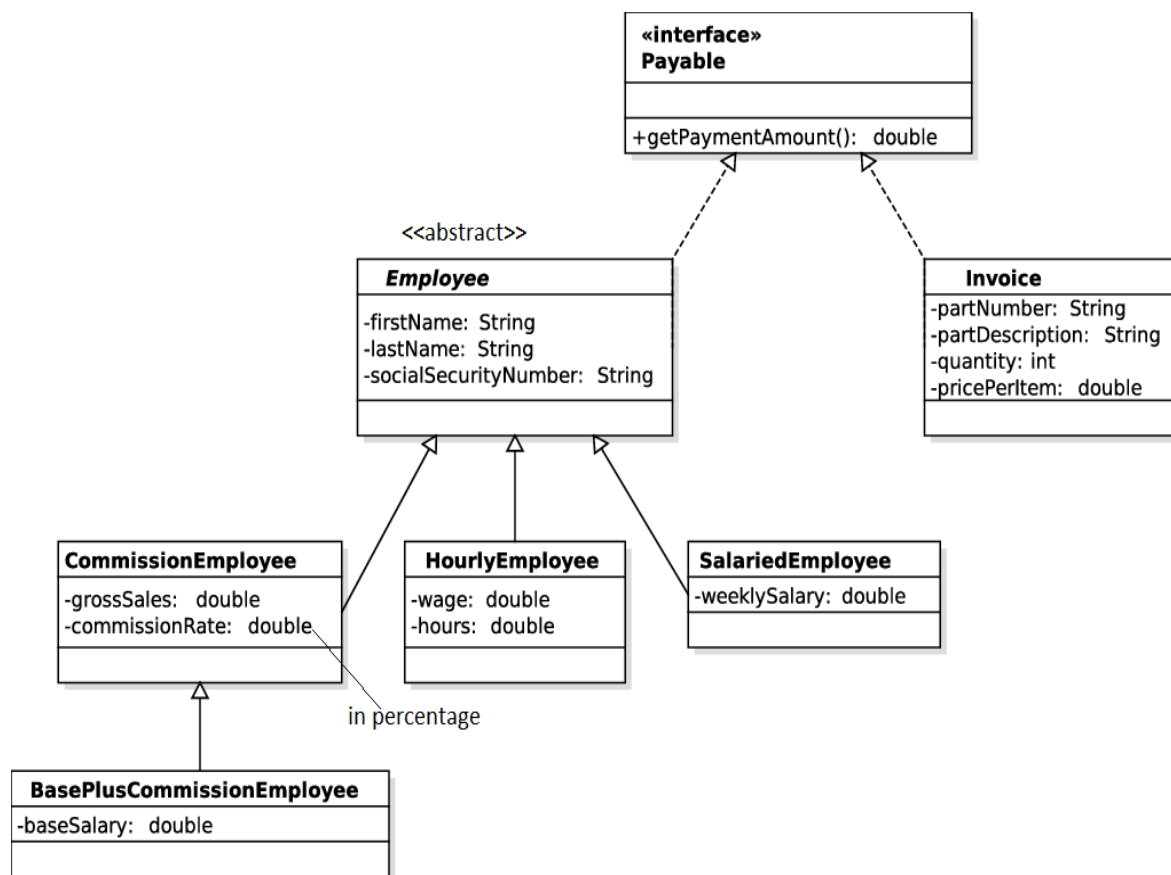


Lesson – 5 – Inheritance and Polymorphism.

Day – 2 – Solve Problem 1 & 2

1. Write a code for the given UML Diagram.

1. Provide necessary getters and setters
2. Provide necessary constructors to initialize values
3. Override the toString() method to display the current status of the objects
4. Write a driver class to test by creating an array of five objects for various employee categories and one object for Invoice class.
5. Apply polymorphism to invoke getPaymentAmount() and display the specific class name(using getClass().getSimpleName()).



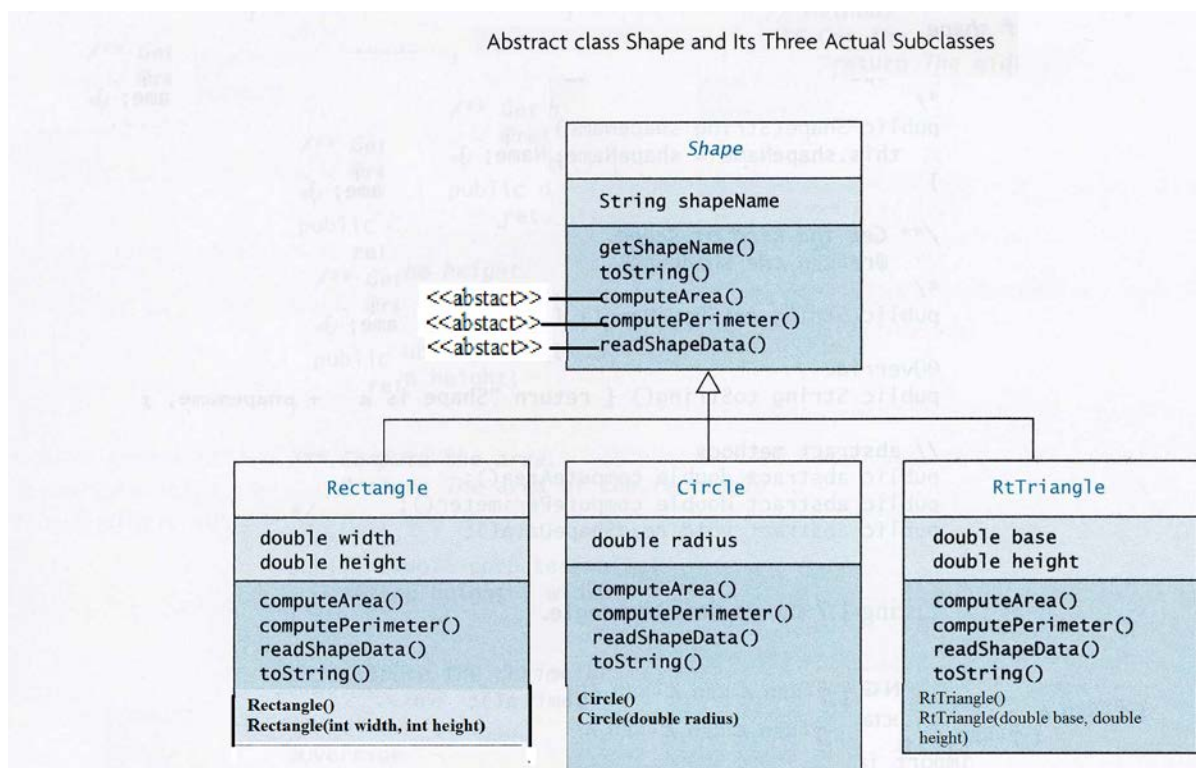
Hints : The `getPaymentAmount()` need to return the values as mentioned below according to the specific class object.

1. **CommissionEmployee** : $\text{grossSales} * \text{CommissionRate}$
2. **BasePlusCommissionEmployee** : $\text{baseSalary} + (\text{grossSales} * \text{CommissionRate})$
3. **HourlyEmployee** : $\text{wage} * \text{hours}$
4. **SalariedEmployee** : weeklySalary
5. **Invoice** : $\text{quantity} * \text{pricePerItem}$

2. Implement the code for the given class diagram.

Problem: We would like to process some standard geometric shapes. Each figure object will be one of three standard shapes (rectangle, circle, right triangle). We would like to be able to do standard computations, such as finding the area and perimeter, for any of these shapes.

- Make Shape class as an Abstract Class
- Create default constructor in each subclass to assign the ShapeName using default constructor and use super by passing the shape name as parameter. (eg : `super("Rectangle");`);
- Create parameter constructor to assign the value of its own instances. (eg: `Rectangle(int width, int height);`);
- Override `toString()` method in all the classes to display the current status of an Object.



Class Rectangle

Data Field	Attribute
<code>double width</code>	Width of a rectangle
<code>double height</code>	Height of a rectangle
Method	Behavior
<code>double computeArea()</code>	Computes the rectangle area ($\text{width} \times \text{height}$)
<code>double computePerimeter()</code>	Computes the rectangle perimeter ($2 \times \text{width} + 2 \times \text{height}$)
<code>void readShapeData()</code>	Reads the width and height
<code>String toString()</code>	Returns a string representing the state

The description of the Rectangle class is presented in the above table. The description of the other classes is similar and need to tries by you.

Perform the following.

1. Create a driver class named as ComputeAreaAndPerimeter.java
2. Introduce a method **public static Shape getShape()** in the main class. It asks the user for the type of figure and returns the instance of the selected choices. The Choices are
 - a. C – for Circle
 - b. R – for Rectangle
 - c. T – for Triangle
3. Through the object return by the getShape() method, invoke readShapeData() – which read the data from the console, then call ComputeArea(), ComputePerimeter() and display the current status of the object.

Note : There are two ways to initialize the values of instances.

1. Using readShapedata() to get the input from the constructors. If you read input through this method create object using default constructor. Eg: Shape Obj = new Rectangle(); Obj.readShapeData()
2. If you want to initialize through constructor use parameterized constructor. Eg: new Rectangle(10,20);

Sample Output :

A sample of the output from ComputeAreaAndPerimeter follows.

```
Enter C for circle
Enter R for Rectangle
Enter T for Right Triangle
R
Enter the width of the Rectangle
120
Enter the height of the Rectangle
200
Shape is a Rectangle: width is 120.0, height is 200.0
The area is 24000.00
The perimeter is 640.00
```

Day – 3

3. Implement the Java code for the given class diagram.

Computer
String manufacturer String processor int ramSize int diskSize double processorSpeed
// Make a Constructor to initialize the instance fields int getRamSize() int getDiskSize() double getProcessorSpeed() double computePower() // return ramSize multiplied by processorSpeed Override the toString() method to display the current status

- Override the equals() and hashCode() method for the Computer class.
- Write a Main class to create an object for Computer to test the operations.
- Compare two Computer Objects are equal or not.

4. Create a class called Person.

```
class Person
{
    String name;
    Computer computer; // Refer from the Problem 3.
}
```

- Apply clone() method and write the implementation to understand the concepts of shallow and deep copy.