**Lab Task 8**

Public interface Shape

{

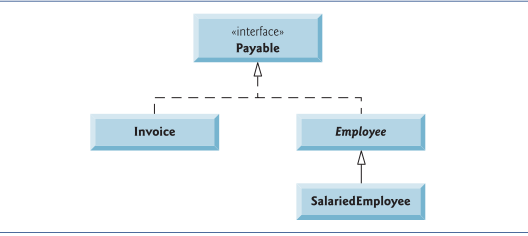
     double getArea();

}

Create two classes Circle and Rectangle. Both must implement the interface Shape.

Note: You can assume appropriate data members for circle and rectangle

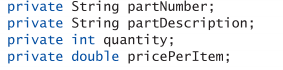
Implement the following hierarchy



Payable:

Double getPaymenyAmount();

Invoice:



Employee:



Salaried Employee:



In the runner , call the getPaymentAmount() method polymorphically.

Create an Encapsulated class Point class with x and y as data members. Create two constructors and a function to move the point. Implement **Cloneable** interface.

Below is the skeleton for a class named “InventoryItem” . Each inventory item has a name and a unique ID number:

class InventoryItem

{

private String name;

private int uniqueItemID;

}

Flesh out the class with appropriate accessors, constructors, and mutatators. This class will implement the following interface:

Public interface compare

{

     boolean compareObjects(Object o);

}

Listed next is a code skeleton for an interface called “Enumeration” and a class called “NameCollection “ . Enumeration provides an interface to sequentially iterate through some type of collection. In this case, the collection will be the class NameCollection that simply stores a collection of names using an array of strings.

interface Enumeration

{

// return true if a value exists in the next index

public boolean hasNext(int index);

// returns the next element in the collection as an Object

public Object getNext(int index);

}

//NameCollection implements a collection of names using a simple array.

class NameCollection

{

String[] names = new String[100];

}

Create constructor and abstract methods of interface in the class NameCollection.

Then write a main method that creates a NamesCollection object with a sample array of strings,

and then iterates through the enumeration outputting each name using the getNext() method.