Chapter 8 lab
For this lab you are required to create the following two class.

1. Point class
2. Coin class

<u>You cannot have any print statements in the Point.java and Coin.java files. All the print statements must be done in the PointTester.java and CoinTester.java</u>

Point class: Write a class encapsulating the concept of a point in the coordinate system. Also write a driver class to test your point class. You must turn in a **Point.java** file for your class and another **PointTester.java** file for the test class.
A point has the following attributes:

1. A double value representing the x coordinate
2. A double representing the y coordinate
3. A class variable count that keeps track of the number of points created

The UML for the point class has been provided. You must implement this class based on the following UML (Unified markup Language)
Description of the methods:

1. Point(): no arguments constructor sets the x and y to zero.
2. Point(double initX, double initY): constructor sets the x to initX and y to initY
3. setX(double newX) and setY(double newY) are mutator methods. They do not return any value.
4. getX() and getY() are accessor methods and they return the values for x and y.
5. getcount(): return the number of points created by the user
6. isOrigion(): determines if a point is the origin or not. If the point is origin it will return true otherwise it will return false.
7. isOnXAxis() determines if the point is located on the x axis, if it is, it will return true, otherwise it will return false.
8. isOnYAxis() determines if a point is on the y axis or not. If it is, it will return true, otherwise it will return false.
9. distanceBetween (Point otherPoint) returns the distance between two points.
10. equals(Point otherPoint) compares two points, if they are the same returns true otherwise it will return false.
11. toString() method that returns a string representing a specific point.

Look at the output for more information.

Here is the UML diagram for the Point class:

Point
Public static count =0 Instance variables - x: double (“-“ means that this instance variable must be private - y: double

"+" means public
+ point()
+point(double initX, double initY)
+setX(double theX): void
+ setY(double theY): void
+ getX( ): double
+ getY( ): double
+getCount():int
+ isOrigin( ): boolean
+ isOnXaxis(): boolean
+ isOnYaxis(): boolean
+ distanceBetween (Point otherPoint): double
+ equals(Point otherPoint): boolean
+ toString( ): String

Here is the shell for Point.java class:

```
public class Point
{
     public static int count =0;
     private double x, y;


   public Point()
   {
       //your code
   }

   public Point(double initX, double initY)
   {
      //your code
   }
   public boolean isOrigin()
   {
     //your code
    }
   public boolean isOnXAxis()
   {
      //your code
   }
   public boolean isOnYAxis()
   {
      //your code
   }
   public double distanceBetween(Point p)
   {
      //your code
   }
   public boolean equals(Point p)
   {
      //your code
   }

   public String toString()
    {
        //your code
    }

    public void setX(double newX)
    {
       //your code
    }
    public void setY(double newY)
    {
       //your code
    }

    public double getX()
    {
       //your code
    }

    public double getY()
    {
       //your code
    }
    public int getCount()
    {
          //your code
    }




     } // end class Point
```

Here is a sample output for the point class:

```
Enter the X value:
¼¼¼¼4
¼¼¼¼Enter the Y value:
```

```
¼¼¼7
¼¼¼(4.0,7.0 ) is not on x axis
¼¼¼(4.0,7.0 ) is on not y axis
¼¼¼Enter the X value:
¼¼¼2
¼¼¼Enter the Y value:
¼¼¼3
¼¼¼Distance between p1(4.0,7.0 )  and p2(2.0,3.0 ) is 4.47
¼¼¼Enter a new  x value of your first point
¼¼¼5
¼¼¼your new point is p1(5.0,7.0 )
¼¼¼p1(5.0,7.0 )  and p2(2.0,3.0 )  are not the same point
¼¼¼p1(5.0,7.0 )  is not on origin
¼¼¼do you have other points to comapre:yes/no
¼¼¼yes
¼¼¼Enter the X value:
¼¼¼5
¼¼¼Enter the Y value:
¼¼¼7
¼¼¼(5.0,7.0 ) is not on x axis
¼¼¼(5.0,7.0 ) is on not y axis
¼¼¼Enter the X value:
¼¼¼8
¼¼¼Enter the Y value:
¼¼¼90
¼¼¼Distance between p1(5.0,7.0 )  and p2(8.0,90.0 ) is 83.05
¼¼¼Enter a new  x value of your first point
¼¼¼9
¼¼¼your new point is p1(9.0,7.0 )
¼¼¼p1(9.0,7.0 )  and p2(8.0,90.0 )  are not the same point
¼¼¼p1(9.0,7.0 )  is not on origin
¼¼¼do you have other points to compare(yes/no)
¼¼¼no
You have created 4 points
```

Coin class

Write a Java class named Coin. You need to turn in the class **Coin.java** and **CoinTester.java**

1. The Coin class should have the following field:
    a. A String named sideUp. The sideUp field will hold either “heads” or “tails” indicating the side of the coin that is facing up.
2. The Coin class should have the following methods:
    i. A no-argument constructor that randomly determines the side of the coin that is facing up (“heads” or “tails”) and initializes the sideUp field accordingly.
    ii. A void method named toss that simulates the tossing of the coin. When the toss method is called, it randomly determines the side of the coin that is facing up (“heads” or “tails”) and sets the sideUp field accordingly.
    iii. A method named getSideUp that returns the value of the sideUp field.
    iv. toString method that return a String representing the coin’s side

3. Write a program(Cointester.java) that demonstrates the Coin class. The program should create an instance of the class and display the side that is initially facing up. Then, use a loop to toss the coin 20 times. Each time the coin is tossed, display the side that is facing up. The program should keep count of the number of times heads is facing up and the number of times tails is facing up, and display those values after the loop finishes.

Coin class
-sideUp : String
+Coin() +getSideUp(): String +toss(): void +toString(): String

Here is the shell for Coin.java class

```
import java.util.Random;
public class Coin
{

   String sideUp;

   public Coin()
   {
     //your code
   }
   public void toss()
   {
      //your code
   }
  public String getSideUp()
   {
      //your code
   }
   public boolean equals(Coin c)
   {
      //your code
  }
   public String toString()
   {
      //your code
   }


}
```

Here is the sample output

```
¼The Tail is up
¼¼¼¼How many times do you want ot toss your coin
¼¼¼¼three
¼¼¼¼Enter an integer value:
¼¼¼¼five
¼¼¼¼Enter an integer value:
¼¼¼¼35
¼¼¼¼Here is the result of tossing your coin 35 times
¼¼¼¼The Tail is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Tail is up
¼¼¼¼The Tail is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Tail is up
¼¼¼¼The Head is up
¼¼¼¼The Tail is up
¼¼¼¼The Head is up
¼¼¼¼The Tail is up
¼¼¼¼The Head is up
¼¼¼¼The Tail is up
¼¼¼¼The Tail is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Head is up
¼¼¼¼The Tail is up
¼¼¼¼The Tail is up
¼¼¼¼The Head is up
```

```
The Tail is up
The Tail is up
The Tail is up
The Head is up
The Head is up
The Head is up
The Head is up
The Tail is up
The Tail is up
The Tail is up
you got 19  Heads and 16 Tails
```