Tutorial #10: More on Classes - SOLUTIONS

Question 1:

Complete the following class definition to represent a point in a 2- dimensional space.

```
public class Point
{
   private int x; // x-coordinate
   private int y; // y-coordinate

   /* A. Write a constructor to set the coordinates of the point to two specific values x1 and y1
   which are passed from the driver. */

   /* B. Write two accessor methods: One to return the content of the x coordinate, the other to
   return the content of the y coordinate. */

   /* C. Write two mutator methods: One to set the content of the x coordinate to some value
   passed from the driver and one to set the y coordinate to some value which again is passed
   from the driver. */

   /* D. Write a method called isEqual which will return true if two points have the same
   coordinates and false otherwise. */

   /* E. Write a method called  reverse which will return a new point with the coordinates
   reversed. That is, if the point which invokes the method has coordinates (a, b), then the
   method should return a new point with coordinates (b, a). */

   /* F. Write a method called  moveBy which will move a point by an integer value. The method
   will add to each coordinate the value passed as an argument. So if the original point was (x1,
   y1), after this method is invoked it will have the coordinates (x1+a, y1+a), where a is the
   argument (the integer value). */

   /* G. Write the toString method such that it displays an object in the following format:
   Coordinates of point are (x, y) where x and y are the contents of the instance variables.
   */

} // end of class Point
```

Answer:

```
// Class definition to represent a point in a 2- dimensional space

public class Point
{
   private int x; // x-coordinate
   private int y; // y-coordinate

 // constructor to set the coordinates of the point
   public Point(int x1, int y1)
   {
      x = x1;
      y = y1;
   }

 // two accessor methods
   public int getX()
   {
      return x;
   }

   public int getY()
   {
      return y;
   }

 // two mutator methods
   public void setX(int x1)
   {
      x = x1;
   }

   public void setY(int y1)
   {
      y = y1;
   }

 /* a method which will return true if two points have the same coordinates
 and false otherwise. */
   public boolean isEqual(Point p)
   {
      return ((x == p.x) && (y == p.y));
   }

 // method called reverse
   public Point reverse ()
   {
      return new Point(y, x);
   }

 // method called moveBy
   public void moveBy(int value)
   {
      x += value;
      y += value;
   }

 // toString method
   public String toString()
   {
      return "Coordinates of point are (" + x + ", " + y+")";
   }
 } // end of class Point
```

Complete the following driver program which tests the class Point defined above.

```
public class PointTest
{
   public static void main(String[] args)
   {
      /* H. Declare 2 points: p1 with coordinates (0,0) and p2 with coordinates (2,3).  */

      /* I. Write the necessary statement(s) to display the coordinates of p1
      and p2.  */

      /* J. Write a statement to reverse the coordinates of p2. */

      /* K. Write the necessary statement(s) to set the coordinates of p1 to be the reverse of
      p2. For example, if p1 is (1,2) and p2 is (2,3) then the coordinates of p1  will be changed to
      (3,2). */

      /* L. Write a statement to add 10 to both coordinates of p1.*/

      /*  M. Write the necessasy statement(s) to display the coordinates of p1 and p2 again.
       */

      /* N. Write the necessary statements to compare the coordinates of points p1 and p2 and
      print “Same” if they have the same coordinates and “Different” if they don’t have the same
      coordinates. */
   }
} // end of class PointTest
```

Answer:

```
public class PointTest
{
   public static void main(String[] args)
   {
      // Declare 2 points
      Point p1 = new Point(0,0);
      Point p2 = new Point(2, 3);

      // display the coordinates of p1 and p2.
      System.out.println(p1);
      System.out.println(p2);

      // reverse the coordinates of p2.
      p2 = p2.reverse() ;

      //set the coordinates of p1 to be the reverse of p2
      p1 = p2.reverse() ;

      // a statement to add 10 to both coordinates of p1.
      p1.moveBy( 10);

      // display the coordinates of p1 and p2 again
      System.out.println();
      System.out.println(p1);
      System.out.println(p2);

      // compare the coordinates of points p1 and p2
      if (p1.isEqual( p2))
      {
         System.out.println("Same");
      }
      else
      {
         System.out.println("Different");
      }
   }
} // end of class PointTest
```

Question 2:

Write a class to represent a temperature. The `Temperature` class should have 2 instance variables:

- a temperature **value** (a double number) and
- a character for the **scale** (either `'C'` for Celsius or `'F'` for Fahrenheit)

The class should have the following methods:

- A no-argument constructor that sets the temperature to zero degrees Celsius.
- A constructor with 2 arguments corresponding to each instance variable. You must check that the character given for the scale is either `'F'` or `'C'`. If it is not the case, then you assign `'C'` to the scale.
- An accessor method called `getTempInCelsius` to return the value of the temperature in Celsius. If the object is stored in Fahrenheit, then the method must convert the temperature using the formula:

  Celsius = (5/9)*(Fahrenheit-32)
- A mutator method to set both the temperature value and the scale. You must check that the character given for the scale is either `'F'` or `'C'`. If it is not the case, then you assign `'C'` to the scale.
- An `isHotter` method to test whether a temperature is hotter (larger) than another. A temperature is considered hotter than another if its value is larger than the other when both temperatures are converted to the same scale (ex. both in Celsius).
- An appropriate `toString` method.
- A method called `add` to add 2 temperatures together and return their sum in Celsius.

Then write a driver program to:

- declare 2 temperature objects: 30 degrees Fahrenheit and 15 degrees Celsius
- call your method `isHotter` and display which temperature is hotter. It is possible that they are the same temperature.
- call your method `getTempInCelsius` and display the first temperature in Celsius
- call your method `add` and display the sum of the two temperatures.

Answer:

```
 /*
 The Temperature class has 2 instance variables: a temperature value (a
 floating point number) and a character for the scale (either 'C' for Celsius
 or 'F' for Fahrenheit) */

public class Temperature
{
   private double value;
   private char scale;

/* A no-argument constructor that sets the temperature to zero degrees
Celsius.*/

   public Temperature()
   {
      value =  0;
      scale = 'C';
   }

/* A constructor with 2 arguments corresponding to each instance variable.*/

   public Temperature(double value, char scale)
   {
      setTemperature(value, scale);
   }

/* An accessor method called getTempInCelsius to return the value of the
temperature in Celsius.*/

   public double getTempInCelsius()
   {
      if (scale == 'C')
         return value;
      else
      {
         //Tc = (5/9)*(Tf-32)
         return (double)5/9 * (value - 32);
      }
   }
```

```
/* A mutator method to set both the temperature value and the scale. */

   public void setTemperature(double value, char Scale)
   {
      this.value = value;
      if (scale == 'F')
         this.scale = scale;
      else
      this.scale = 'C';
   }

/* An isHotter method to test whether a temperature is hotter (larger) than
another.*/

   public boolean isHotter(Temperature another)
   {
      return this.getTempInCelsius() > another.getTempInCelsius();
   }

//An appropriate toString method.
   public String toString ()
   {
      return "temperature " + value + " " + scale;
   }

/* A method called add to add 2 temperatures together and return their sum in
Celsius.*/

   public double add(Temperature another)
   {
      return this.getTempInCelsius() +
      another.getTempInCelsius() ;
   }
} // End of class Temperature


public class TemperatureDriver
{
   public static void main(String[] args)
   {
/*declare 2 temperature objects: 30 degrees Fahrenheit and 15 degrees
Celsius*/

      Temperature temp1 = new Temperature(30, 'F');
      Temperature temp2 = new Temperature(15, 'C');
/* call your method isHotter and display which temperature is hotter */

     if (temp1.isHotter(temp2))
        System.out.println(temp1 + " is hotter than " + temp2);
     else if(temp2.isHotter(temp1))
        System.out.println(temp2 + " is hotter than " + temp1);
     else
        System.out.println("The two temperatures are the same");

/* call your method getTempInCelsius and display the first temperature in
Celsius */

     System.out.println("The value in Celcius of " + temp1 + " is "
        + temp1.getTempInCelsius() );

/* call your method add and display the sum of the two temperatures. */

     System.out.println("The sum in Celcius of " + temp1 + " and "
        + temp2 + " is " + temp1.add( temp2) );

   }//End of main
}  //End of TemperatureDriver
```