1. Show how you would:
   a. Declare an array of integers called foo

   ```java
   int[] foo;
   ```

   b. Create storage for 13 integers referenced by foo

   ```java
   foo = new int[13];
   ```

   c. Fill the array with the numbers 10 through 22 (using a for loop)

   ```java
   for(int i = 0; i < foo.length; i++)
     foo[i] = i + 10;
   ```

2. Given an array of doubles called `temps` that contains the average daily temperatures over the past month, describe what the following code fragment does and what it computes.

   ```java
   double sum = 0;
   for(int i = 0; i < temps.length; i++) {
     sum += temps[i];
   }
   sum = sum / temps.length;
   ```

   **This code adds up all of the values in the array and divides them by the length of the array, thereby computing the average of the values in the array**

   **This code computes the average temperature over the past month**
3. Suppose that you are given a class called Circle that has two methods:
   • void setDiameter(double diameter) - set the diameter of the circle
   • double getArea() - get the area of the circle

  a. Show how you would declare a Circle object called myCircle, create the object, and set its diameter to 7

```java
Circle myCircle;
myCircle = new Circle();
myCircle.setDiameter(7);
```

b. Show how you would get and print out the area of this circle.

```java
System.out.println(myCircle.getArea());
```

4. Show how you would implement a class called Point that stores an (x,y) pair of values representing a location in 2D space and has the following methods:
   • set() - takes two doubles as parameters and stores them as the x and y values of the Point
   • getX() - gets the x value
   • getY() - gets the y value
   • add() - takes a Point as a parameter, adds it to the current Point, and returns a new Point representing the sum of the two

```java
class Point {
    double x,y;
    
    public void set(double a, double b) {
        x = a;
        y = b;
    }
    public double getX() {
        return x;
    }
    public double getY() {
        return y;
    }
    public Point add(Point other) {
        Point newPoint = new Point();
        newPoint.set(x + other.x, y + other.y);
        return newPoint;
    }
}
```