1. Write a method named `sumSquares` with two parameters `a, b` of type `double` that returns the sum of squares `a^2 + b^2` of its two parameters.

```java
public double sumSquares(double a, double b) {
    double result = a * a + b * b;
    return result;
}
```

2. Write a mutator (setter) method for a class with an instance variable `age` of type `int`.

```java
public void setAge(int a) {
    age = a;
}
```

3. Write a constructor for a class named `Emoji`. Assume the class has only one instance variable named `code`, of type `int`. The constructor should have no parameters and should set `code` to zero.

```java
public Emoji() {
    code = 0;
}
```

4. Write another constructor the class `Emoji`, above. This version should have a parameter (of appropriate type) and it should set `code` to the value given by that parameter.

```java
public Emoji(int c) {
    code = c;
}
```

(See back of this page for Circle.java class to use as reminder.)
import java.awt.*;

public class Circle
{
    private int diameter, x, y;
    private Color color;

    public Circle(int size, Color shade, int upperX, int upperY)
    {
        diameter = size;
        color = shade;
        x = upperX;
        y = upperY;
    }

    public void draw(Graphics page)
    {
        page.setColor(color);
        page.fillOval(x, y, diameter, diameter);
    }

    public void setColor(Color shade)
    {
        color = shade;
    }

    public Color getColor()
    {
        return color;
    }

    public String toString()
    {
        return "O";
    }
}
1. Write a method called `cube` that accepts one integer parameter and returns that value raised to the third power. (Note: compute the cube by multiplying, e.g., \(x^3\) or use `Math.pow(x,y)`).

   ```java
   public int cube(int x)
   {
       double result = x * x * x;
       return result;
   }
   ```

2. Write an accessor (getter) method for a class with an instance variable `interest` of type `double`.

   ```java
   public double getInterest()
   {
       return interest;
   }
   ```

3. Suppose you have a class named `Emoji` with a single instance variable `code` of type `int`. Write a `toString()` method that returns a text-based representation of the emoji corresponding to the `code`. For simplicity, we only represent 3 codes
   - 0 = sad,
   - 1 = happy,
   - 2 = wink.

   For example, the “wink” Emoji can be represented by the String “;(-)”

   ```java
   public String toString()
   {
       String message = "";
       if (state == 0)
           message = ":-)";
       else if (state == 1)
           message = ":-(";
       else
           message = ";(-)";
       return message;
   }
   ```

(See back of this page for Circle.java class to use as reminder.)
import java.awt.*;

public class Circle
{
    private int diameter, x, y;
    private Color color;

    // Constructor: Sets up this circle with the specified values.
    public Circle(int size, Color shade, int upperX, int upperY)
    {
        diameter = size;
        color = shade;
        x = upperX;
        y = upperY;
    }

    // Draws this circle in the specified graphics context.
    public void draw(Graphics page)
    {
        page.setColor(color);
        page.fillOval(x, y, diameter, diameter);
    }

    // Color mutator.
    public void setColor(Color shade)
    {
        color = shade;
    }

    // Color accessor.
    public Color getColor()
    {
        return color;
    }

    // toString
    public String toString()
    {
        return “O”;
    }
}