Variables and Designing a Java Class

Your team has been tasked with designing a simple Circle class. After some back and forth, your agree that Circle objects should be able to do the following:

- create new circle given a center point and a radius
- calculate the area
- calculate the perimeter
- given a point, determine if the point is inside the circle
- shift the circle to have a new center
- scale the radius
- give center
- give radius

You may assume the existence of the Point class below as you work on your Circle class definition.

```
public class Point {
  private int x;
  private int y;

  public Point (int x, int y) {
    this.x = x;
    this.y = y;
  }

  public int getX() {
    return x;
  }

  public int getY() {
    return y;
  }

  public void shift(int xChange, int yChange) {
    x += xChange;
    y += yChange;
  }
}
```

Design Questions

As a team, discuss and come to a consensus on your answers to the questions below.

1. What data fields does the Circle class need to have?

2. What values should be passed to the constructor so that it can initialize the fields of a newly created Circle object?

3. Which items in the list of Circle functionality given above describe accessors?

4. Decide on names for your accessors, then write the complete definition for each method.

5. The value of π is needed to calculate the area and perimeter of a circle. Of the four possible kinds of Java variables, what should π be?

6. Write the **signatures only** for the methods needed to complete the Circle class functionality. Once you are done, compare your method signatures with those of another team. Is there disagreement on any of the following?

   a. method names
   b. parameter names
   c. return values

   Together as a class: report out on agreements/disagreements and decide on the final signature for each Circle class method.

7. **At the board**, work out the implementation for each Circle class method. When you are done discuss the following:

   - *Could* the variables for the circle's center and radius be declared final? *Should* they be declared final?
   - Your constructor can reuse the provided center point, or create a new point of its own. Discuss the advantages and disadvantages of each approach.
   - If the Point class did not have a shift method, would this cause any problems for your Circle class implementation?
   - Suppose that instead of changing the fields of the existing object, calling shift on a Point returned a new Point at the shifted location. How would your Circle class implementation need to change?
   - Suppose that instead of changing the existing object, the shift and rescale methods for the Circle class are changed so that they return a new Circle object.
     - Could the fields for the Circle class be changed to *final*?
     - What would be the implications of this change, assuming it is possible?

   Together as a class: report out and discuss answers to questions.