

# Java Programming I: Exercise 6

*The deadline for submissions is one week after the exercise*

The aim of this exercise is to learn the concept of inheritance hierarchy. You will also investigate different strategies to debug programs.

Imagine that you are a member of the software development team. You are designing the Java code to help your junior students understand key features of inheritance. There are three problems to solve.

## [Core Set, Problem 1] A Hierarchy of Classes

(4 points)

You should create Java code to implement a hierarchy of classes (See Figure 1).

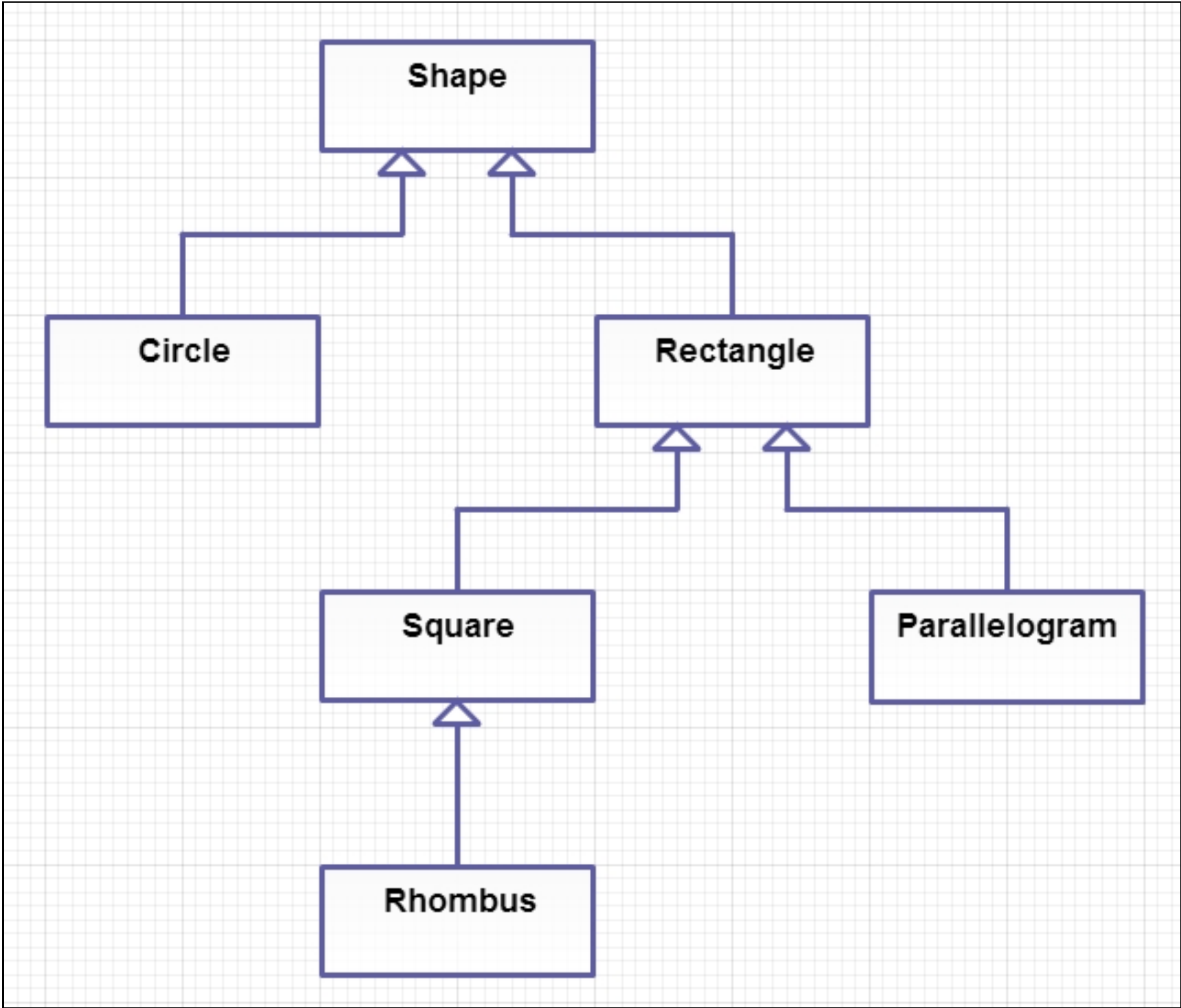


Figure 1: A Hierarchy of Classes

Code for the Shape class is as follows:

```
class Shape {
    private String name;
    public Shape(String name) {
```

```
        this.name = name;
    }
    public Shape() {
        name = "no name";
    }
    public String getName() {
        return name;
    }
    public String toString() {
        return getClass().getName() + " " + name;
    }
}
```

Instant data for each class should be as follows:

- *Circle*: a radius (cm)
- *Rectangle*: adjacent sides (cm)
- *Square*: a side (cm)
- *Parallelogram*: adjacent sides and a diagonal (see Figure 2) (cm)
- *Rhombus*: a side and a diagonal (see Figure 2) (cm)

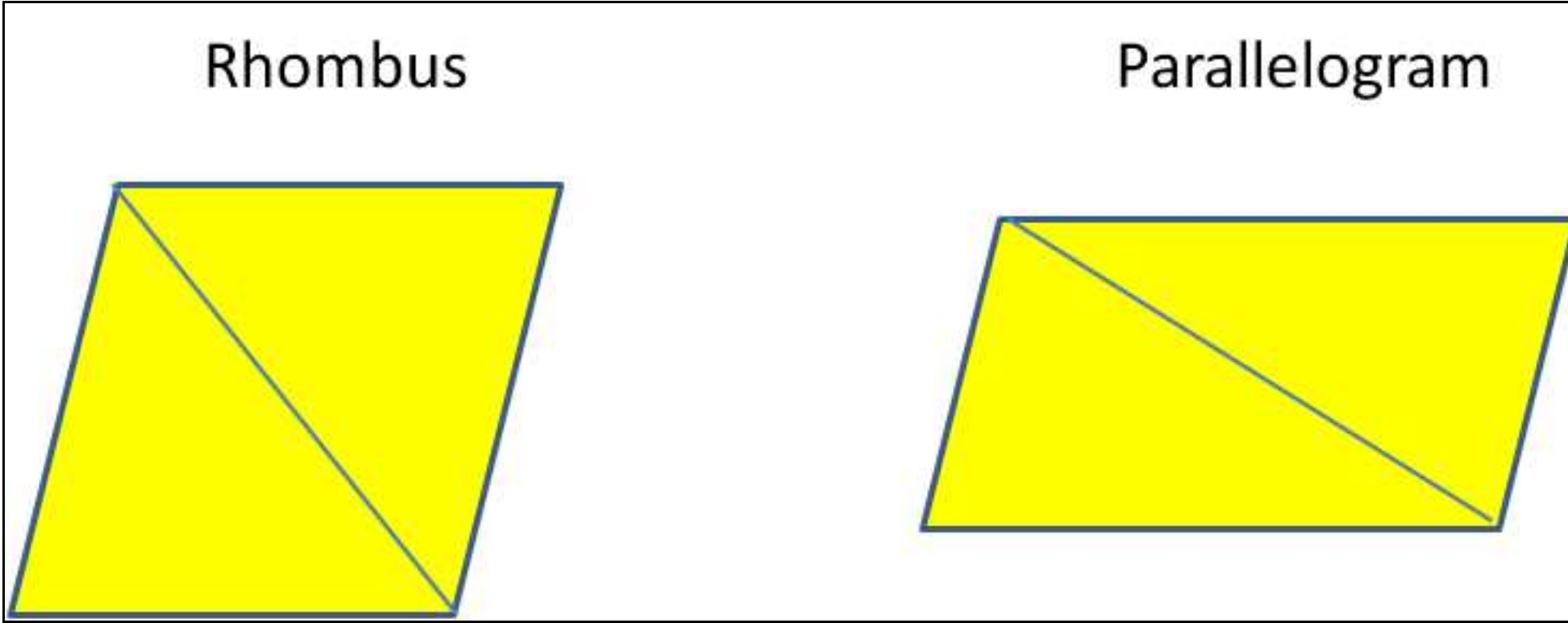


Figure 2: Rhombus and Parallelogram

You should define a constructor for each class to accept and initialize instant data. You should override the *toString* method in each class. You should include a method that calculates and returns the perimeter of the figure. Another method should return the area of the figure.

You should create a *TestFigures* class, whose main method instantiates each figure. The method should print instant data, the area and the perimeter of each figure.

Your Java code should be saved to one file (.java).

## [Core Set, Problem 2] A Hierarchy of Classes: Variant Two

(6 points)

Analyze the hierarchy of classes presented in Figure 1. Propose another hierarchy. Apply your hierarchy when follow the description of Problem 1. Add the *equals* method to each class. Use Command Line Debugger jdb discussed at the lecture to debug your code.

# [Advanced Set, Problem 3] The Case of the Confusing Constructor

(12 points)

The code below presents you with two Confusing constructors. The main method invokes a constructor, but which one? There are several errors in the source code. You should debug this code. Use Command Line Debugger jdb discussed at the lecture. Fixed code should print one line. What does the program print? You should explain the reason for the output of the program.

The file name for your answers is *Confusing.pdf*.

```
class Confusing {
    private Confusing (Object o) { // DO NOT CHANGE THIS LINE
        Systemout.println("Object");
    }
    private Confusing (double[] dArray) { // DO NOT CHANGE THIS LINE
        System.out.println("double array");
    }
    static void main(String args) {
        o = new Confusing(null); // DO NOT CHANGE THIS LINE
    }
}
```