

QAP2: JAVA
Mohammad Hossain
SD11

Instructions:

You are allowed to complete the assessment problems below in whatever way you can but please answer the following questions/points as part of your submission:

1. How many hours did it take you to complete this assessment? (Please keep try to keep track of how many hours you have spent working on each individual part of this assessment as best you can - an estimation is fine; we just want a rough idea.) ~ 6 hours

2. What online resources you have used? (My lectures, YouTube, Stack overflow etc.)
~ Lectures and YouTube

3. Did you need to ask any of your friends in solving the problems. (If yes, please mention name of the friend. They must be amongst your class fellows.)
~ N/A

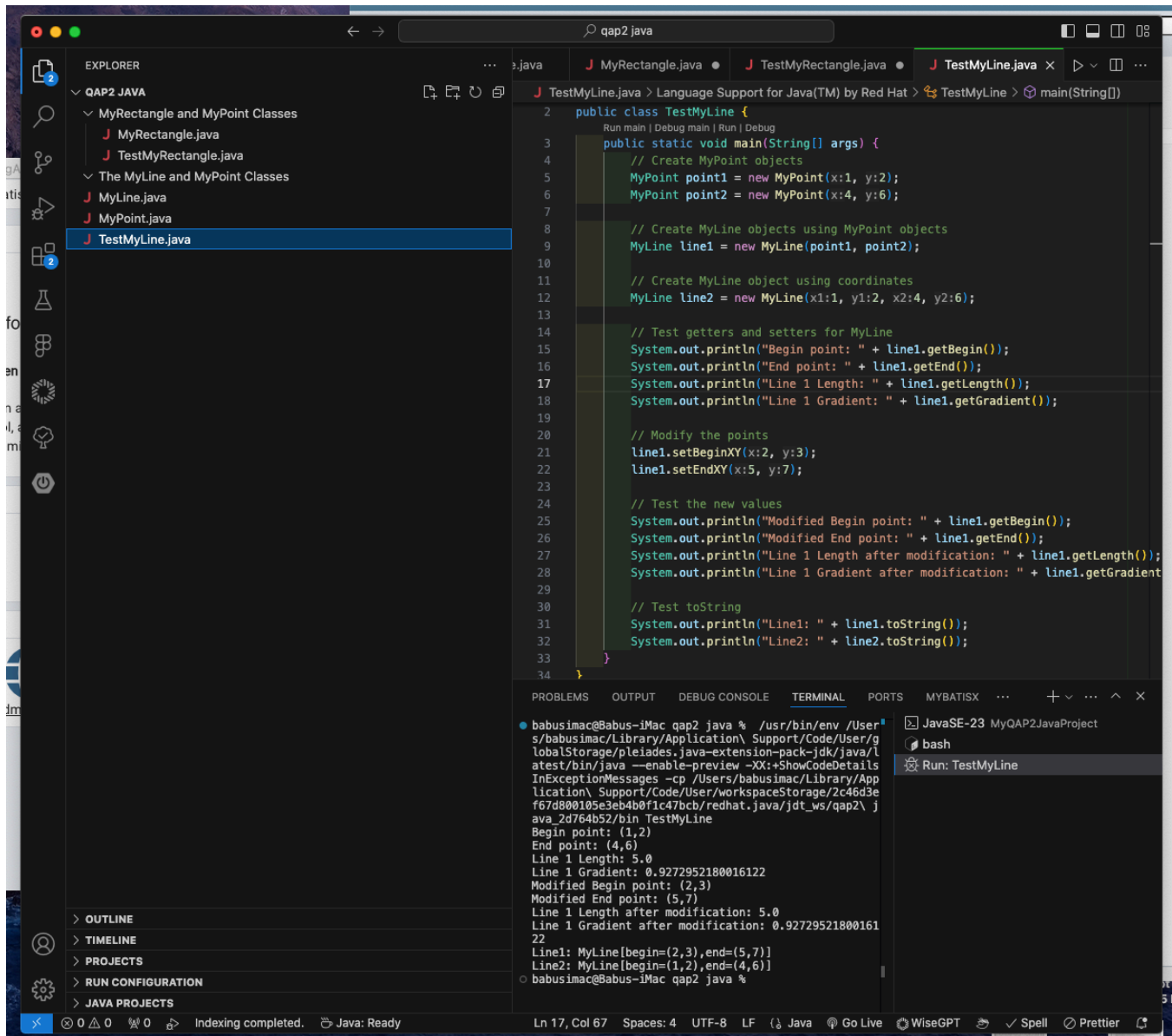
4. Did you need to ask questions to any of your instructors? If so, how many questions did you ask (or how many help sessions did you require)?
~ N/A

5. Rate (subjectively) the difficulty of each question from your own perspective, and whether you feel confident that you can solve a similar but different problem requiring some of the same techniques in the future now that you've completed this one.
~ Medium difficulty. Confident to solve similar problems with same techniques.

Thanks.

Problem 1:

Screenshots of output:



The screenshot displays an IDE interface with the following components:

- EXPLORER:** Shows a project named 'QAP2 JAVA' with subfolders 'MyRectangle and MyPoint Classes' and 'The MyLine and MyPoint Classes'. Files include 'MyRectangle.java', 'TestMyRectangle.java', 'MyLine.java', 'MyPoint.java', and 'TestMyLine.java'.
- Editor:** Displays the code for 'TestMyLine.java'. The code defines a 'MyLine' class with methods for creating points, lines, and testing getters/setters. It also includes a 'main' method that demonstrates these operations.
- TERMINAL:** Shows the command 'qap2 java %' and the resulting output, which prints the details of the created lines and points.

```
public class TestMyLine {  
    public static void main(String[] args) {  
        // Create MyPoint objects  
        MyPoint point1 = new MyPoint(x:1, y:2);  
        MyPoint point2 = new MyPoint(x:4, y:6);  
  
        // Create MyLine objects using MyPoint objects  
        MyLine line1 = new MyLine(point1, point2);  
  
        // Create MyLine object using coordinates  
        MyLine line2 = new MyLine(x1:1, y1:2, x2:4, y2:6);  
  
        // Test getters and setters for MyLine  
        System.out.println("Begin point: " + line1.getBegin());  
        System.out.println("End point: " + line1.getEnd());  
        System.out.println("Line 1 Length: " + line1.getLength());  
        System.out.println("Line 1 Gradient: " + line1.getGradient());  
  
        // Modify the points  
        line1.setBeginXY(x:2, y:3);  
        line1.setEndXY(x:5, y:7);  
  
        // Test the new values  
        System.out.println("Modified Begin point: " + line1.getBegin());  
        System.out.println("Modified End point: " + line1.getEnd());  
        System.out.println("Line 1 Length after modification: " + line1.getLength());  
        System.out.println("Line 1 Gradient after modification: " + line1.getGradient());  
  
        // Test toString  
        System.out.println("Line1: " + line1.toString());  
        System.out.println("Line2: " + line2.toString());  
    }  
}
```

Output:

```
Begin point: (1,2)  
End point: (4,6)  
Line 1 Length: 5.0  
Line 1 Gradient: 0.9272952180016122  
Modified Begin point: (2,3)  
Modified End point: (5,7)  
Line 1 Length after modification: 5.0  
Line 1 Gradient after modification: 0.9272952180016122  
Line1: MyLine[begin=(2,3),end=(5,7)]  
Line2: MyLine[begin=(1,2),end=(4,6)]
```

Problem 2:

Diagram:

Class Diagram, MyRectangle

```
-----  
- topLeft: MyPoint  
- bottomRight: MyPoint  
-----  
+ MyRectangle(x1: int, y1: int, x2: int, y2: int)  
+ MyRectangle(topLeft: MyPoint, bottomRight: MyPoint)  
+ getTopLeft(): MyPoint  
+ setTopLeft(topLeft: MyPoint): void  
+ getBottomRight(): MyPoint  
+ setBottomRight(bottomRight: MyPoint): void  
+ getWidth(): int  
+ getHeight(): int  
+ getArea(): int  
+ getPerimeter(): int  
+ toString(): String
```

Class Diagram, MyPoint:

```
-----  
- x: int  
- y: int  
-----  
+ MyPoint(x: int, y: int)  
+ getX(): int  
+ setX(x: int): void  
+ getY(): int  
+ setY(y: int): void  
+ toString(): String
```

Screenshot of output:

The screenshot shows an IDE with a dark theme. The Explorer panel on the left lists files under 'QAP2 JAVA' and 'The MyLine and MyPoint Classes'. The main editor displays the source code of 'TestMyRectangle.class', which is a decompiled version of a Java class. The code defines a 'TestMyRectangle' class with a constructor and a 'main' method. The 'main' method creates two 'MyPoint' objects and two 'MyRectangle' objects, then prints their dimensions, area, and perimeter. The bottom panel shows the output of running the code, which matches the printed statements in the code. The status bar at the bottom indicates 'Ln 1, Col 1' and 'Spaces: 3'.

```
// Source code is decompiled from a .class file using FernFlower decompiler.
public class TestMyRectangle {
    public TestMyRectangle() {
    }

    public static void main(String[] var0) {
        MyPoint var1 = new MyPoint(1, 5);
        MyPoint var2 = new MyPoint(6, 1);
        MyRectangle var3 = new MyRectangle(var1, var2);
        MyRectangle var4 = new MyRectangle(2, 7, 8, 3);
        System.out.println("Rectangle1: " + String.valueOf(var3));
        System.out.println("Width of Rectangle1: " + var3.getWidth());
        System.out.println("Height of Rectangle1: " + var3.getHeight());
        System.out.println("Area of Rectangle1: " + var3.getArea());
        System.out.println("Perimeter of Rectangle1: " + var3.getPerimeter());
        System.out.println("Rectangle2: " + String.valueOf(var4));
        System.out.println("Width of Rectangle2: " + var4.getWidth());
        System.out.println("Height of Rectangle2: " + var4.getHeight());
        System.out.println("Area of Rectangle2: " + var4.getArea());
        System.out.println("Perimeter of Rectangle2: " + var4.getPerimeter());
    }
}
```

babusimac@Babus-iMac qap2 java % javac MyPoint.java MyRectangle.java TestMyRectangle.java

babusimac@Babus-iMac qap2 java % java TestMyRectangle

Rectangle1: MyRectangle[topLeft=(1,5), bottomRight=(6,1), width=5, height=4]
Width of Rectangle1: 5
Height of Rectangle1: 4
Area of Rectangle1: 20
Perimeter of Rectangle1: 18
Rectangle2: MyRectangle[topLeft=(2,7), bottomRight=(8,3), width=6, height=4]
Width of Rectangle2: 6
Height of Rectangle2: 4
Area of Rectangle2: 24
Perimeter of Rectangle2: 20

babusimac@Babus-iMac qap2 java %

Problem 3:

Screenshot of Output:

