Problem Set 3, Part I

Please start your answer to each problem on a new page, as we have done below!

Problem 1: Our `Rectangle` class revisited

1-1)

```
type of method: mutator
header:
public void rotate() {
}
```

1-2)

```
type of method: accessor
header:
public boolean largerThan(Rectangle other) {
}
```

1-3)

```
problems in code:
We cannot call r1.width and r1.height directly because the fields are
private by encapsulation.

rewritten version:
Rectangle r1 = new Rectangle(60, 80);
System.out.println("r1's height is: " + r1.getHeight());
r1.grow(20, 0);
System.out.println(r1);
```

Problem 2: A class that needs your help

2-1) *Revise the code found below:*

```
public class ValuePair {
    private int a;
    private double b;

    public double product() {
        return this.a * this.b;
    }

    // add the new methods here
    public int getA() {
        return this.a;
    }

    public double getB() {
        return this.b;
    }

    public void setA(int a) {
        if (a % 2 == 0) {
            throw new IllegalArgumentException();
        }
        this.a = a;
    }

    public void setB(double b) {
        if (b <= 0.0) {
            throw new IllegalArgumentException();
        }
        this.b = b;
    }

    public ValuePair(int a, double b) {
        this.setA(a);
        this.setB(b);
    }
}
```

2-2)

a) `ValuePair vp1 = new ValuePair(7, 15.5);`
b) `vp1.setA(25);`
c) `double b1 = vp1.getB();`
d) `vp2.setA(vp2.getA() + 2);`

Problem 3: Static vs. non-static

3-1)

type and name of the variable	**static or non-static?**	**purpose of the variable, and why it needs to be static or non-static**
double rawScore	non-static	stores the raw score associated with a given Grade object; needs to be non-static so every Grade object will have its own instance of this variable
String category	non-static	stores the category associated with a given Grade object; needs to be non-static so every Grade object will have its own category of “assignment”, “quiz”, or “exam”
int numAssignment	static	builds an integer to count the number of Grade object in “assignment” category; needs to be static so it belongs to Grade Class as a whole
int numQuiz	static	builds an integer to count the number of Grade object in “quiz” category; needs to be static so it belongs to Grade Class as a whole
int numExam	static	builds an integer to count the number of Grade object in “exam” category; needs to be static so it belongs to Grade Class as a whole

3-2)

a) static or non-static?: non-static
explanation: The method needs to access to the fields of a particular object and change its own category.

b) changes it would need to make: The method will change the non-static variable category from “quiz” into “exam”. The static variable numQuiz will reduce by 1, and the static variable numExam will increase by 1.

3-3)

a) static or non-static?:non-static
explanation: The method needs to access the fields of a particular object and change its own latePenalty.

b) example of calling it: g.waiveLatePenalty();

3-4)

a) static or non-static?: static

explanation: The method does not need a called method because all of the information it needs is provided by its parameters pct and possiblePoints.

b) example of calling it: Grade.computeRaw(pct,possiblePoints);