

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:accessor

Header: public void boolean(Rectangle height,Rectangle width)

1-2)

type of method:mutator

Header:public void int(Rectangle width, Rectangle height)

1-3)

problems in code: the instance variables are trying to be directly accessed even though they are private. Also, the rectangle that should be created should use the RectangleClient, not rectangle (I think)

rewritten version:

```
Rectangle r1 = new Rectangle(30, 50);
System.out.println(r1.getWidth());
r1.setHeight() = r1.getHeight() * 2;
System.out.println(r1);    // print the new dimensions
```

Problem 2: A class that needs your help

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

```
public class StreetAddress {
    private int num;
    private String name;

    public int getNum(){
        return this.num;
    }

    public String getName(){
        return this.name;
    }
    public void setName(String name){

        if(name >= 30){
            throw new IllegalArgumentException
        }
        else{
            this.name = name;
        }

    }
    public int getNum(int num){

        if(num <= 0){
            throw new IllegalArgumentException
        }
        else{
            this.num = num;
        }

    }
    public StreetAddress(int num, String name){
        this.name = name;
        this.num = num;
    }

    public boolean onEvenSide() {
        return (this.num % 2 == 0);
    }

}
```

2-2)

```
a) StreetAddress ccds = new StreetAddress(650, "Commonwealth Avenue");  
b) ccds.setNum(665);  
c) String streetName = ccds.getName();  
d) addr.setNum(addr.getNum()+1);
```

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
int rank	non-static	stores the rank (1, 2, 3, ...) associated with a given Card object; needs to be non-static so every Card object will have its own instance of this variable
private value"	non-static	Stores the value of each card, the value is not static, because every card will have a different value based on the game played
private suit	non-static	Stores the value of each suite, there are four different suits, so they can't all have the same value, so it will not be the same for each object
private heartCount;	static	Stores how many objects have been initialized
private spadeCount;	static	Stores how many objects have been initialized
private diamondCount;	static	Stores how many objects have been initialized
private clubsCount;	static	Stores how many objects have been initialized

3-2)

a) static or non-static?:non-static

explanation: each object has an associated suit, and each the suit variable is a private instance variable. This means, that in order to access the variable and make the necessary, changes the methods needs to be static.

b) changes it would need to make: value, suit,, rank, decrease hearCount, and increase spadeCount

c) example of calling it:c.setSuit();

3-3)

a) static or non-static?:

explanation: static

b) example of calling it:Card.numFaceCards();