Paper Compiling with Java Objects name: _____

Directions: For each exercise, carefully digest the Java code. Model each object using the samples from the online tutorial (i.e. draw a box, label the class name, member vars, and methods). Then <u>compute by hand</u> exactly what will show up in the NetBeans output when this program is executed. Write your answer in the box at the bottom of each exercise.

PaperObjects 1 : EBook

// scratch space

```
class EBook{
    public String title;
    public int numPages;
    public int numPagesRead;
    public double minutesReading;
}

public class PaperObjects1 {

    public static void main(String[] args) {

        EBook javaBook = new EBook();

        EBook pythonBook = new EBook();


        javaBook.title = "Thinking In Java";

        javaBook.numPages = 950;

        javaBook.numPagesRead = 400;


        pythonBook.numPages = 1400;

        pythonBook.numPagesRead = 12;

        pythonBook.minutesReading = 111.3;


        int delta = javaBook.numPages - pythonBook.numPages;

        System.out.println(delta);

        javaBook.numPages = pythonBook.numPages;

        delta = javaBook.numPages - pythonBook.numPages;

        System.out.println(delta);

        System.out.println(pythonBook.title);

        System.out.println(javaBook.title.toUpperCase());


    } // close main

} // close class PaperObjects1
```

<u>Hand-computed program output:</u>

Paper Objects 2: Battery

```
class Battery{
    private final double POWER_USED_PER_MIN = 0.25;
    private double powerRemaining = 400.0;
    public double useBattery(double time){
        double PwrUsed = time * POWER_USED_PER_MIN;
        powerRemaining = powerRemaining - PwrUsed;
        return powerRemaining;
    }
    public void charge(){
       powerRemaining = powerRemaining + 200;
    }
    public double getPowerRemaining(){
        return powerRemaining;
    }
 }
public class PaperObjects2 {
    public static void main(String[] args) {
        Battery energizer; Battery duracell;
        energizer = new Battery();
        duracell = new Battery();

        System.out.println("Energizer start: " + energizer.getPowerRemaining());
        System.out.println("Duracell start: " + duracell.getPowerRemaining());

        energizer.useBattery(100);
        duracell.useBattery(1000);
        energizer.useBattery(800);
        energizer.charge();
        Battery batt3 = duracell;
        batt3.charge();

        System.out.println("Energize end: " + energizer.getPowerRemaining());
        System.out.println("Energize end: " + duracell.getPowerRemaining());
        System.out.println("Energize end: " + batt3.getPowerRemaining());
    } // close main
} // close class
```

// scratch space

Hand-computed program output:

Paper Objects 3:

```
class Plane{
    final int MAX_DEGREES = 360;
    private int heading = 0;  private int numPass = 0;
    public void changeHeaing(int newHeading){
        if(newHeading < MAX_DEGREES && newHeading >= 0){
            heading = newHeading;
        } else {
            System.out.println("ERROR: impossible heading!");
        }
    }
    public int getHeading(){
        return heading;
    }
    public int getNumPass(){
        return numPass;
    }
    public void setNumPass(int num){
        numPass = num;
    }
}
```

Hand-computed program output:

```
public class PaperObjects4 {
    public static void main(String[] args) {
        Plane superJet = new Plane();
        superJet.setNumPass(100);
        superJet.changeHeaing(50);
        System.out.println("Heading 1: " + superJet.getHeading() + " deg");
        superJet.changeHeaing(360);
        System.out.println("Heading 2: " + superJet.getHeading() + " deg");
        superJet.setNumPass(-12);
        superJet.changeHeaing(-20);
        System.out.println("Heading 3: " + superJet.getHeading() + " deg");
        System.out.println("Passegers: " + superJet.getNumPass());
    }
}
```

Quash that bug!

Directions: The following code contains four erros in LOGIC that MAY prevent the class from compiling. Find the three lines with errors, CORRECT/quash them, and compute the program's output after the changes.

```
class PersianRug{

    public double length;

    public double width;

    public void shrinkRung(double perc){

        System.out.println("Shrinking by " + perc + "%");

        length = length - (length * (perc / 100.0));

        width = width * (perc / 100.0);

    }

}

public class QuashThatBug1 {

    public static void main(String[] args) {

        PersianRug coolRug = new PersianRug();

        length = 100.0;

        width = 800.0;

        coolRug.shrinkRung(20.0);

        System.out.println("Length: " + coolRug.getLength());

        System.out.println("Width: " + coolRug.setWidth());

    }

}
```

Hand-computed program output after bug fixes are complete: