

2. Lab Grades

Suppose your lab instructor has a somewhat complicated method of determining your grade on a lab. Each lab consists of two out-of-class activities—a pre-lab assignment and a post-lab assignment—plus the in-class activities. The in-class work is 60% of the lab grade and the out-of-class work is 40% of the lab grade. Each component of the grade is based on a different number of points (and this varies from lab to lab)—for example, the pre-lab may be graded on a basis of 20 points (so a student may earn 17 out of 20 points) whereas the post-lab is graded on a basis of 30 points and the in-class 25 points. To determine the out-of-class grade the instructor takes the total points earned (pre plus post) divided by the maximum possible number of points, multiplied by 100 to convert to percent; the in-class grade is just the number of points earned divided by the maximum points, again converted to percent.

The attached program *LabGrade.java* is supposed to compute the lab grade for a student. To do this it gets as input:

- The number of points the student earned on the pre-lab assignment and the maximum number of points the student could have earned,
- The number of points earned on in-class work and the maximum number of points, and
- The number of points earned on the post-lab assignment and the maximum number of points.

The lab grade is computed as described above: the in-class and out-of-class grades (in percent) are computed separately then a weighted average of these is computed.

1. First carefully hand trace the program assuming the input stream contains the values 17, 20, 23, 25, 12, 15. Trace the program exactly as it is written (it is not correct but it will compile and run so the computer would not know it isn't correct).
 - a. Show exactly how the computer would execute the assignment statement that computes the out of class average for this set of input. Show how the expression will be evaluated (the order in which the operations are performed) and what the result will be.
 - b. Show how the computer would execute the assignment statement that computes the in-class average. What will the result be?
 - c. Show how the computer would execute the assignment statement that computes the lab grade.
2. Now run the program, typing in the input you used in your trace. Compare your answers to the output. Clearly the output is incorrect! Correct the program. This involves writing the expressions to do calculations correctly. The correct answers for the given input should be :

Out of class average: 82.857 (the student earned 29 points out of a possible 35)
In-class average of 92 (23 points out of 25)
Lab grade of 88.34 (40% of 82.857 plus 60% of 92).

3. Modify the program to make the weights for the two components of the grade variable rather than the constants 0.4 and 0.6. To do this, you need to do four things:
 - a. Change the declarations so the weights (*IN_WEIGHT* and *OUT_WEIGHT*) are variables rather than constants. Note that you should also change their names from all capital letters (the convention for constants) to lowercase letters with capitals starting new words (the convention for variables). So *IN_WEIGHT* should become *inWeight*. Of course, you'll also have to change it where it's used in the program.
 - b. In the input section, add statements that will prompt the user for the weight (in decimal form—for example .4 for 40%) to be assigned to the in-class work, then read the input. Note that your prompt should explain to the user that the weight is expected to be in decimal form.

- c. In the section that calculates the labGrade add an assignment statement that calculates the weight to be assigned to the out of class work (this will be 1 minus the in-class weight).

Compile and run your program to make sure it is correct.

```
// *****  
// LabGrade.java  
// This program computes a student's lab grade from  
// the grades on the three components of lab: the pre-lab  
// assignment, the lab itself, and the post-lab assignment.  
// *****  
  
import java.util.Scanner;  
  
public class LabGrade  
{  
    public static void main (String[] args)  
    {  
        // Declare constants  
        final double IN_WEIGHT = 0.6; // in-class weight is 60%  
        final double OUT_WEIGHT = 0.4; // out-of-class weight is 40%  
  
        // Declare variables  
        int preLabPts; //number of points earned on the pre-lab assignment  
        int preLabMax; //maximum number of points possible for pre-lab  
        int labPts; //number of points earned on the lab  
        int labMax; //maximum number of points possible for lab  
        int postLabPts; //number of points earned on the post-lab assignment  
        int postLabMax; //maximum number of points possible for the post-lab  
        int outClassAvg; //average on the out of class (pre and post) work  
        int inClassAvg; //average on the in-class work  
        double labGrade; //final lab grade  
  
        Scanner scan = new Scanner(System.in);  
  
        // Get the input  
        System.out.println("\nWelcome to the Lab Grade Calculator\n");  
        System.out.print("Enter the earned points on the pre-lab: ");  
        preLabPts = scan.nextInt();  
        System.out.print("What was the maximum number of points for pre-lab? ");  
        preLabMax = scan.nextInt();  
        System.out.print("Enter the earned points on the post-lab: ");  
        postLabPts = scan.nextInt();  
        System.out.print("What was the maximum number of points for the post-lab? ");  
        postLabMax = scan.nextInt();  
        System.out.print("Enter the earned points on the lab: ");  
        labPts = scan.nextInt();  
        System.out.print("What was the maximum number of points for the lab? ");  
        labMax = scan.nextInt();  
        System.out.println();  
  
        // Calculate the average for the out of class work  
        outClassAvg = preLabPts + postLabPts / preLabMax + postLabMax * 100;  
  
        // Calculate the average for the in-class work  
        inClassAvg = labPts / labMax * 100;  
  
        // Calculate the weighted average taking 40% of the out-of-class average  
        // plus 60% of the in-class
```

```
labGrade = IN_WEIGHT * outClassAvg + OUT_WEIGHT * inClassAvg;

// Print the results
System.out.println("Your average on out-of-class work is " + outClassAvg + "%");
System.out.println("Your average on in-class work is " + inClassAvg + "%");
System.out.println("Your lab grade is " + labGrade + "%");
System.out.println();
    }
}
```

Deliverables

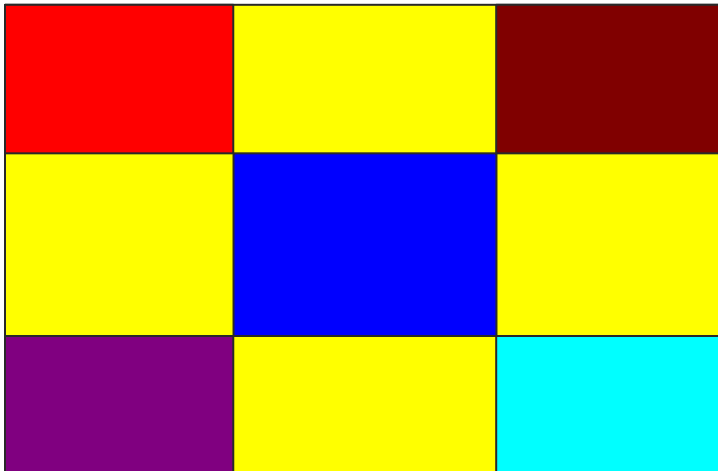
- A list of answers for question 1.
- A printout of the corrected program and the final execution from question 2.
- A printout of the modified program and the final execution from question 3.

3. The Java Coordinate System

The Java coordinate system is discussed in Section 2.7 & 2.9 of the text. Under this system, the upper left-hand corner of the window is the point (0, 0). The X axis goes across the window, and the Y axis goes down the window. So the bigger the X value, the farther a point is to the right. The bigger the Y value, the farther it is down. There are no negative X or Y values in the Java coordinate system. Actually, you can use negative values, but since they're off the screen they won't show up!

1. Save the attached files *Coords.java* and *Coords.html* to your local directory. File *Coords.java* contains an applet that draws a rectangle whose upper left-hand corner is at 0,0. Run this applet through a web browser.
2. Modify the applet so that instead of 0,0 for the upper left-hand corner, you use the coordinates of the middle of the applet window. This applet is set up to be 600 pixels wide and 400 pixels high, so you can figure out where the middle is. Save, compile, and view your applet. Does the rectangle appear to be in the middle of the screen? Modify the coordinates so that it does appear to be in the middle. In order to see the changes in the web browser, click “refresh” button with control key.
3. Now add four more rectangles to the applet, one in each corner. Each rectangle should just touch one corner of the center rectangle and should go exactly to the edges of the window.
4. Make each rectangle be a different color. To do this, use the `setColor` method of the `Graphics` class to change the color (this is already done once). **Do not change the background color once it has been set!** Doing so causes the screen to flicker between colors.

Example screen:



```
// *****  
//   Coords.java  
//  
//   Draw rectangles to illustrate the Java coordinate system  
//  
// *****  
  
import javax.swing.JApplet;  
import java.awt.*;  
  
public class Coords extends JApplet  
{
```

```

public void paint (Graphics page)
{
    // Declare size constants
    final int MAX_SIZE = 300;
    final int PAGE_WIDTH = 600;
    final int PAGE_HEIGHT = 400;

    // Declare variables
    int x, y;    // x and y coordinates of upper left-corner of each shape
    int width, height; // width and height of each shape

    // Set the background color
    setBackground (Color.yellow);

    // Set the color for the next shape to be drawn
    page.setColor (Color.blue);

    // Assign the corner point and width and height
    x = 0;
    y = 0;

    width = 150;
    height = 100;

    page.fillRect(x, y, width, height);

}
}

```

Coords.html

```

<HTML>
  <HEAD>
    <TITLE> Java Coordinate System </TITLE>
  </HEAD>

  <BODY BGCOLOR="lightgreen">
    <H1 align="center"> Java Coordinate System </H1>

    <P>This web site demonstrates the Java coordinate system!</P>
    <applet code="Coords.class" width=600 height=400>
    </applet>
  </BODY>
</HTML>

```

Deliverables

- Screenshots from question 1, 2, 3, and 4.
- Printouts of the program from question 2, 3, and 4.

PROJECT 1 DELIVERABLES

Submit **hardcopy** and **softcopy to lab assistant**.

- A cover page with the project number, due date, and the names of your Project Team Members.
- Deliverables from the exercise 1, 2, and 3.
- This page, with the appropriate signature and date, indicating that the project has been completely and correctly demonstrated in lab.

<p>LABORATORY SIGNATURE</p> <p>PROJECT TEAM MEMBERS:</p> <p>STUDENT NAME _____</p> <p>STUDENT NAME _____</p> <p>STUDENT NAME _____</p> <p>_____</p> <p>LAB INSTRUCTOR SIGNATURE DATE _____</p>
--