MAT 2170: Laboratory 3

Key Concepts

The purpose of this lab is to familiarize you with arithmetic expressions in Java.

1. Primitive Data Types `int` and `double`
2. Operations involving primitive data types, especially integer division
3. Constant declaration and usage
4. Variable declaration and usage
5. Valid identifiers

Notes

- Complete Worksheet #3 and be prepared to show it to your instructor during lab.
- For this week's Lab exercises, create each project in a `mat2170/lab3` directory under your user account (i.e., not in any other directory). This can be done at the time you create each project — especially the first one. The group project should be completed, then shared with all members so they can publish it as well.
- When you have completed all exercises:
  1. Make sure you have published all of these programs to your web page
  2. Submit an electronic copy of this lab.
  3. Print each program (group program only required once), staple in the order assigned, and hand in at the beginning of lab 4.
  4. Complete Worksheet #4 by the beginning of lab 4.

Exercises

1. (Exer. 3, Pg 91) Design and implement a Java dialog program, `Savings`, that reads in two floating point numbers: an **account balance** and an **annual interest rate** expressed as a *percentage*. Display the balance at the end of one and again at the end of two years — assuming no deposits or withdrawals — just the interest payment. Note that the interest should be compounded annually, i.e., added to the bank balance at the end of each year (and thus used for the beginning balance the following year).

   > To display multiple lines of output in a single dialog box, use one or more `print()` statements followed by a `println()`. The formatting character "\n" (backslash-n) can be used to force a new line of output within the same dialog box.

2. Design and implement a Java dialog program, `CylinderVolume`, which asks the user for the **radius** of a cylinder and its **height**, then computes the volume of that cylinder using the formula:

   $$\texttt{Volume} = \pi \times \texttt{radius}^2 \times \texttt{height}$$

   There is no "raise to a power" operator in Java, so for now, you will need to explicitly square the radius. Create and use a `constant` for π with the value `3.14159`.

3. Design and implement a Java dialog program, `DistanceSum`, that will compute the sum of two distances, in yards, feet, and inches. The two distances will be entered by the user (also in yards, feet, and inches). When you report the total distance, be sure that the number of feet is less than three and the number of inches less than twelve. The answer should be displayed in a single dialog box.

Continues on back.

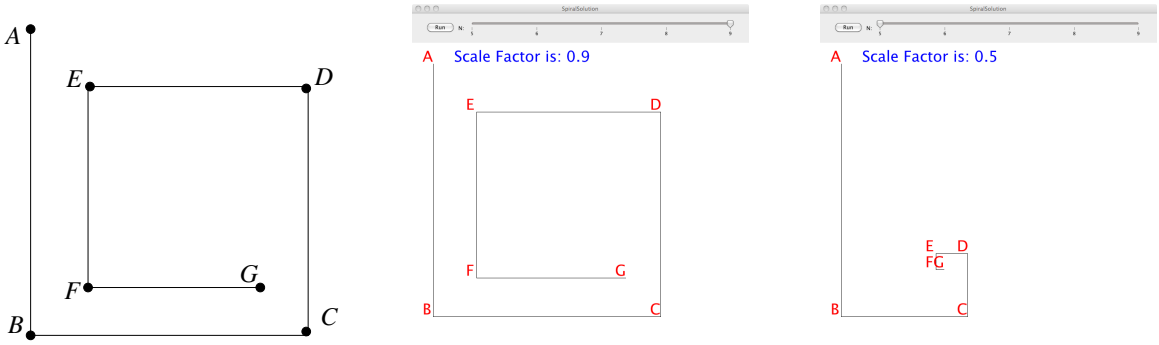
Keep in mind the following guidelines:

- Be sure to include your name and the purpose of your program in the opening comments, and use line comments and appropriate constants.
- Prompt the user separately for the yards, feet, and inches for each of the distances you need.
- Your program should convert both distances into inches, add, and then display the sum in the correct format. Declare constant `int` objects which are used to assist with the distance conversions. For example, `InchesPerFoot`, `FeetPerYard`, and `InchesPerYard` would be three useful constants.
- A sample of what your program should produce is shown below.

```
Calculating the sum of two distances
------------------------------------

Enter the first distance:
Yards:  2
Feet:   2
Inches: 8

Enter the second distance:
Yards:  5
Feet:   1
Inches: 6

The total distance is 8 yards, 1 feet, and 2 inches.
```

- If the user enters 2 yards, 2 feet, and 8 inches, followed by 5 yards, 1 foot, and 6 inches, the total distance is 8 yards, 1 feet and 2 inches.

4. (Group Project) Design and implement a Java SliderProgram, `Spiral`, which calculates the coordinates of a spiral based on an initial length of 600 and a scaling percentage based on the slider, then draws and labels the spiral.



The first diagram (on the left, above) shows a spiral. Each "leg" of the spiral gets shorter by a fixed percentage as you move toward the center. In the **example** shown, the length of BC is 90% of AB, the length of CD is 90% of BC, and so on. In the second picture, the slider has been set to 9 (translated to 90%), and in the last picture, the slider was set to 5 (translated to 50%).

The first endpoint (A) appears at the point `<MARGIN, MARGIN>` of an 800×800 window. The **length of the initial side**, AB, is 600 pixels, and the left and top margins are 50 pixels (given as a constant in the skeleton) are *givens*. Use the slider to find the **scaling percentage**. The skeleton program has the statements to correctly size the window, get the slider value and begin modifying it, and has a constant declared for the margin. Complete the program by turning the scaling percentage into an equivalent decimal, determine the coordinates of the seven spiral corners, then drawing the spiral. The letter labels can be placed by offsetting the `GLabel` -25 in the x coordinate, and -5 in the y coordinate of the associated endpoint. The label with the scale factor can be placed at `MARGIN` + 50 for the x and `MARGIN` - 5 for the y coordinates. I used the font `"SansSerif-36"`.