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

Exercises

Complete the Prelab exercises and be prepared to show them to your instructor during lab.

For the following 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.

1. (Exer. 3, Pg 91) Design and implement a Java dialog program, `Savings`, that reads in two numbers: an **account balance** and an **annual interest rate** expressed as a *percentage*. Display the balance at the end of one and 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. (Exer. 4, Pg 92) Design and implement a Java dialog program, `CircleArea`, which asks the user for the **radius** of a circle, then computes the area of that circle using the formula:

   $$\texttt{Area} = \pi \times \texttt{radius}^2$$

   There is no "raise to a power" operator in Java, so for now, you will need to explicitly square the radius. Use `3.14159` as the value for π, which should be created as a `constant`.

3. Design and implement a Java dialog program, `ElapsedTime`, that will compute the elapsed time, in hours, minutes, and seconds between a starting time and a finishing time (also given in hours, minutes, and seconds) that have been entered by the user. (See an example on the back of this page.)

   Keep in mind the following guidelines:

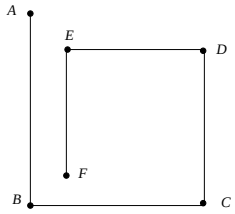
   - The overall form of the program should be modeled on the lab programs. Be sure to include your name(s) and the purpose of your program in the opening comments, and use line comments and appropriate constants.
   - Enter times using a 24-hour system.
   - Prompt the user separately for the hours, minutes, and seconds for each of the times you need.
   - You may assume that the finishing time is later than the starting time (but within the same day).
   - Your program should convert both times into seconds, subtract, and then display the difference in the correct format. Declare constant `int` objects which are used to assist with the time conversions. For example, `SecondsPerMinute`, `MinutesPerHour`, and `SecondsPerHour` would be three useful constants.

- A sample of what your program should produce is shown below.

```
Calculating the difference between two times
-------------------------------------------

Enter the initial time:
Hour:   6
Minute: 48
Second: 23

Enter the final time:
Hour:   19
Minute: 31
Second: 52

The elapsed time is 12 hours, 43 minutes, and 29 seconds.
```

4. Design and implement a Java dialog program, `SpiralCoords`, which calculates the coordinates of a spiral based on the initial length and scaling percentage entered by the user. (Note: this is *not* a graphics–based program; you are *not* being asked to draw the spiral.) You are encouraged to create a few test cases to help you solve this problem and ensure your program is correct.

   The diagram below 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.



   The point A appears at the origin of an arbitrarily large window. Given (1) the **length of the initial side**, AB, and (2) the **scaling percentage**, your program is to determine the coordinates of points B through F. Your Java program should:

   - prompt for and obtain the **length** of the initial side
   - prompt for and obtain the **scaling percentage**
   - calculate and display the **coordinates** of the six points (use `print` rather than `println`, so all coordinates show up neatly in one dialog box)

   When you have completed all exercises:

   (a) Print each program and staple to the Postlab.

   (b) Submit an electronic copy of your work for this lab by dragging your `lab3` folder to the submit icon on your dock, then choose `2170`.

   (c) Publish all of these programs to your web page, listed under `Lab 3` as `Savings`, `CircleArea`, `ElapsedTime`, and `SpiralCoords`. To do this, create `html` files, update your `index.html` file, and copy the `jar` files to `www` directory. Then open an `iTerm` terminal window and give the command: `websync`

   (d) Complete the Postlab worksheet, staple your printouts to it, and hand in at the beginning of lab 4.

Lab 3— Prelab Name: _____

1. Complete the test suite for the external documentation for the `Savings` program. Select amounts and interest rates that will test the correctness of your program. Determine the correct answers using a calculator.

   Problem Statement: *Given a dollar amount and annual interest rate from the user, determine the balance of a savings account at the end of one and two years, assuming annual compounding of interest.*

   Specifications:

   (a) INPUT: `Amount` — `double`, non–negative; `InterestRate` — `double`, non–negative

   (b) OUTPUT: `Balance_1` — `double`, non–negative; `Balance_2` — `double`, non–negative

   Test Suite:

Amount	Interest Rate	Year 1	Year 2
6000	4.25	6255.0	6520.8375
10000	6.75		

   General Algorithm:

   - Prompt for and get data
   - Calculate and display balance for first year
   - Calculate and display balance for second year

   Detailed Algorithm:

   - Prompt for and get data
     - Prompt for and get Amount
     - Prompt for and get Interest Rate
     - Change Interest Rate into decimal equivalent
   - Calculate and display balance for first year
     - Determine Interest for first year, add to Amount
     - Display balance after first year
   - Calculate and display balance for second year
     - Determine Interest for second year (based on updated value), add to Amount
     - Display balance after second year

Continued on back

2. Complete the external documentation for the `CircleArea` program by filling in the Problem Statement, Test Suite table, and Detailed Algorithm. Select various radii to test the correctness of your program. Use a calculator to determine the areas in the test suite.

   (a) **Problem Statement**:

   (b) **Specifications**:

      i. INPUT: `Radius` — `double`, non–negative

      ii. OUTPUT: `Area` — `double`, non–negative

   (c) **Test Suite**:

Radius	**Area**
`10`	`314.159265`

   (d) **General Algorithm**:

   - Prompt for and get Radius
   - Calculate Area
   - Display Area

   (e) **Detailed Algorithm**: (*Since this is a very simple program, just give the Java statements you plan to use.*)

   - Prompt for and get Radius

   - Calculate Area

   - Display Area

Lab 3— Postlab Name: _____

Hand this sheet in at the beginning of Lab 4.
You may need to consult the textbook in order to answer the following:

1. What are the two attributes that define a data type?

   (a)

   (b)

2. Circle each of the following which represent legal Java variable names:

a)	`x`	e)	`short`	i)	`12MonthTotal`
b)	`formula1`	f)	`tiny`	j)	`margin-cost`
c)	`average_rainfall`	g)	`total output`	k)	`b4hand`
d)	`%correct`	h)	`aReasonablyLongName`	l)	`_stk_depth`

3. Indicate the **value** and **type** — ***i*** (int) or ***d*** (double) — for each of the following expressions:

Expression	Value	Type
`2 + 3`		
`19.0 / 5`		
`19 % 5`		
`19 / 5`		
`3 * 6.0`		
`2 % 7`		

4. By applying the appropriate precedence rules, evaluate the following expressions:

   (a) `6 + 5 / 4 - 3` ☐

   (b) `6 + (5 / 4) - 3` ☐

   (c) `6.0 + 5 / 4 - 3` ☐

   (d) `6 + 5.6 / 4 - 3` ☐

   (e) `2 + 2 * (2 * 2 - 2) % 2 / 2` ☐

   (f) `10 + 9 * ((8 + 7) % 6) + 5 * 4 % 3 * 2 + 1` ☐

   (g) `1 + 2 + (3 + 4) * ((5 * 6 % 7 * 8) - 9) - 10` ☐

Continued on back

5. Complete the following table. The object `MyNumber` is of type `int`.

`MyNumber`	`(MyNumber + 5)/6`
1	
2	
5	
6	
7	

`MyNumber`	`(MyNumber + 5)/6`
11	
12	
16	
18	
28	

6. Suppose you are putting empty beverage bottles into six-packs in order to return them to the local recycling center. If you have `MyNumber` bottles to recycle, what does the quantity `(MyNumber + 5)/6` represent? Write your answer as a simple English statement involving the *number of bottles* and the *number of six-pack containers* involved.

7. Using a calculator, determine the number of hours, minutes, and seconds we can obtain from the total seconds given:

Total Seconds	**Hours**	**Minutes**	**Seconds**
302	0	5	2
11002			
72			
60			
61			
7380			
37394			