Project: Variables and Expressions

Deliverables

Your project files should be submitted to Web-CAT by the due date and time specified. In order to avoid a late penalty for the project, you must submit your completed code files to Web-CAT by 11:59 p.m. on the due date. If you are unable to submit via Web-CAT, you should e-mail your project Java files in a zip file to your TA before the deadline.

Files to submit to Web-CAT:

- MidpointOfLineSegment.java
- TimeConverter.java

Specifications

Overview: You will write two programs this week. One will determine the midpoint of a line segment given the x and y coordinates for the two end points of the line segment, and the other will calculate the number of days, hours, minutes, and seconds from a raw measurement of seconds.

- **MidpointOfLineSegment.java**

Requirements: A program is needed that takes in the x and y coordinates for the two endpoints of a line segment, and then calculates the midpoint of the line segment using the following formula.

$$midPoint = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right)$$

where the coordinates of the two end points are (x_1, y_1) and (x_2, y_2) respectively.

Design: The main method of the program should read in the coordinates for the first point and store them in variables (e.g., x1 and y1) of type double, read in the coordinates for the second point and store them in variables (e.g., x2 and y2) of type double, and then use the formula above to compute and print the x and y coordinates for the midpoint. The examples below show the program output for various input values.

Example #1

```
Line #  Program output
1       Enter the coordinates for endpoint 1:
2          x1 = 0
3          y1 = 0
4       Enter the coordinates for endpoint 2:
5          x2 = 4
6          y2 = 4
7
8       For endpoints (0.0, 0.0) and (4.0, 4.0), the midpoint is (2.0, 2.0).
```

Note that lines 2, 3, 5, and 6 begin with tab.

Example #2

```
Line #  Program output
1       Enter the coordinates for endpoint 1:
2          x1 = -1.5
3          y1 = -2.5
4       Enter the coordinates for endpoint 2:
5          x2 = 3.5
6          y2 = 7.5
7
8       For endpoints (-1.5, -2.5) and (3.5, 7.5), the midpoint is (1.0, 2.5).
```

Example #3

```
Line #  Program output
1       Enter the coordinates for endpoint 1:
2          x1 = .25
3          y1 = 2.5
4       Enter the coordinates for endpoint 2:
5          x2 = -1.75
6          y2 = 5.5
7
8       For endpoints (0.25, 2.5) and (-1.75, 5.5), the midpoint is (-0.75, 4.0).
```

Code: Create a Scanner object on System.in to read in the values for the coordinates. The values of type double should be read in using the Scanner nextDouble method.

Test: You are responsible for testing your program, and it is important to not rely only on the examples above. Remember that the input values for the x and y coordinates are doubles, so be sure to test both positive and negative values (with and without a decimal point). You should use a calculator or jGRASP interactions to check your answers. You may also find it useful to graph the line segment.

- **TimeConverter.java**

Requirements: A digital timer manufacturer would like a program that accepts a raw time measurement in seconds (of type int) and then then displays the time as a combination of days, hours, minutes, and seconds. When a negative raw time measurement is entered, an appropriate message is printed as shown in the first of the two examples below.

Design: The digital timer manufacturer would like the output to look as shown below when the test values -1234 is entered as the raw time for one run and 1234567 is entered for another run.

Line #	Program output
1	`Enter the raw time measurement in seconds: -1234`
2	`Measurement must be non-negative!`

Line #	Program output
1	`Enter the raw time measurement in seconds: 1234567`
2	
3	`Measurement by combined days, hours, minutes, seconds:`
4	`   days: 14`
5	`   hours: 6`
6	`   minutes: 56`
7	`   seconds: 7`
8	
9	`1234567 seconds = 14 days, 6 hours, 56 minutes, 7 seconds`

Your program must follow the above format with respect to the output. Note that lines 4 through 7 for the amount 1234567 begin with tab (i.e., your output should use the **escape sequence for a tab**).

Code: In order to receive full credit for this assignment, you must calculate the number of days, hours, minutes, and seconds and store each of the values in separate variables. Create a Scanner object on System.in to read in the value for the raw time using the nextInt() method. It is recommended as a practice that you do not modify input values once they are read in and stored.

Test: You will be responsible for testing your program, and it is important to not rely only on the example above. Assume that the amount entered can be any integer less than or equal to 2,147,483,647 (the maximum value for a 32 bit int) and greater than or equal to -2,147,483,648 (the minimum value for a 32 bit int).

Grading

Web-CAT Submission: You must submit both "completed" programs to Web-CAT at the same time. Prior to submitting, be sure that your programs are working correctly and that have passed Checkstyle. **If you do not submit both programs at once, the submission will receive zero points for correctness**. Activity 1 describes how to create a jGRASP project containing both of your files.