Due: Monday, September 3, 2012 by 11:59 PM

Deliverables

The following project files must be uploaded to Web-CAT by the due date and time specified above (see the Lab Guidelines for information on submitting project files). A portion of the lab is due on Wednesday in lab. If the Wednesday portion is not complete, there will be a 5% deduction on your project grade. **Projects sent via e-mail past the deadline at 11:59 PM will not be accepted without a university-approved excuse.**

Files to submit to Web-CAT:

- TableOrder.java
- DistanceCalculator.java

Specifications

Overview: You will write two programs this week. One will calculate the number of tables and plates needed for a catered dinner, and the other will computer the distance and midpoint between two points given their X and Y coordinates.

- **TableOrder.java**

  Requirements: A caterer would like a program that will allow the user to enter the name of an event and the number of people attending. In return, the program should display the name of the event **in double quotes**, the number of tables that will be needed and the number of plates that will be needed (see the Design section). A table holds 10 people. The number of plates should be the number of guests plus 15 (in case some plates get broken).

  Design: The caterer would like the output to look as shown below (replace everything in *italics* with your own words):

Line number	Program output
1	*Prompt user for name of event*: `Grand Opening`
2	*Prompt user for the number of guests*: `10`
3	
4	`Tables for "Grand Opening":`
5	`        Tables needed: 1`
6	`        Plates needed: 25`

Your program must follow the above format with respect to the output. Note that lines 5 and 6 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 plates in the println statement rather than being stored as a variable (5% of program specifications). It is recommended as a practice that you do not modify input values once they are 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 number of people can be any number greater than or equal to 0 (if no one is attending, no tables are needed).

- **DistanceCalculator.java**

Requirements: A program is needed that will take the x and y coordinates of two points as integers, calculate the midpoint between the two points, and calculate the slope of the line segment defined by the two points.

Design: Your program output should be as follows in regard to line numbers (replace everything in *italics* with your own words):

Line number	Program output
1	*Prompt user for X and Y of first point*`:`
2	` x1 = 4`
3	` y1 = 6`
4	*Prompt user for X and Y of second point*`:`
5	` x2 = 6`
6	` y2 = 8`
7	`The midpoint is (5.0, 7.0)`
8	`The slope is 1.0`

Note that lines 2, 3, 6, and 7 begin with tab.

The equation for finding the midpoint and slope are as follows (where x1≠ x2):

$$midpoint = \left(\frac{x1 + x2}{2}, \frac{y1 + y2}{2}\right)$$

$$slope = \frac{y2 - y1}{x2 - x1}$$

Code: Use an if-else statement to determine if x1 = x2. If true, print "`The slope is undefined`" on line 8; if false, print value of the slope as shown in the example above. You are allowed to use only **4 variables of type int** for each of the X and Y coordinates **and 1 variable of type double** for the slope. Casting may be required to ensure correct results in expressions and assignment statements. The calculations for *midpoint* should be done with expressions in the print statement. Using any more variables (or using any of the above variables for other calculations) will result in a loss of 5% for project specifications.

Test: You will be responsible for testing your program, and it is important to not rely only on the example above. Remember that the midpoint and slope calculations both result in doubles, so be sure to test values for the X and Y coordinates that result in non-whole numbers as well.

Grading

Web-CAT Submission: You should submit your programs to the “Ungraded” assignment in Web-CAT to make sure you have your programs named correctly. After this is successful, you'll have 10 tries to submit your programs to the “Graded” assignment in Web-CAT. However, you should test your program thoroughly and submit it with the intent of getting full credit on the first submission. **Make sure that you submit both programs at once, or the submission will receive zero points for correctness**. Activity 1 describes how to create a jGRASP project containing both of your files.