Lab 22 (November 27 or November 28)

Instructions: Complete the exercises below. Be sure to show your code to one of the lab TAs before you leave, so that you can receive credit for this lab. You must also upload a copy of all your source code (`.java`) files through the link on Blackboard by 11:59 PM on Tuesday, November 28.

1. Write a small Java program that implements a very simple arithmetic calculator for integer values (you should support the five basic operations: addition, subtraction, multiplication, division, and modulus). Your program should prompt the user to enter (on a single line, separated by whitespace) the two operands and the operation to be performed. It then prints the result on a new line. For example:

   Please enter a calculation to perform: **1 + 2**
   3

   Your program should include an exception-handling mechanism (a `try-catch` block) that deals with illegal operations (basically, division by 0 or an unrecognized arithmetic operator; you may assume that the user will always enter integer values for the operands). In the event that your program raises an exception, your program should display a message that informs the user of the type of error before it exits (without printing any type of answer):

   Please enter a calculation to perform: **1 / 0**
   Division by zero is illegal

   or

   Please enter a calculation to perform: **4 $ 22**
   $ is an invalid operator

   Recall that you can declare an `Exception` with a `String` that describes the details of a particular abnormal situation, e.g., `new Exception("Too few input values!")`.

2. Suppose that a plain text file named "test.txt" contains an unspecified number of integer scores. Write a small Java program to read the contents of this file and display their total (an integer) and average (as a `double`). Scores are separated by blanks. There are at least 50 scores in the file.

   Note: If you are using Eclipse (which you most likely are), data files should be placed directly in the project folder, **NOT** inside the "bin" or "src" subfolders. For example, if your project was named "Lab 22 Q1", your data file would go in *that* folder, not in (for example) "Lab 22 Q1/src". If necessary, you can drag and drop files inside Eclipse's Package Explorer to move your data file outside of the "src" folder, if it accidentally ends up in that directory.

Sample test.txt file contents:

```
1 2 3 4 5 6 7 8 9 0 8 7 65 5 4 3 4 6 7 8 76 5 4 6 8 98 76 5 43
6 78 76 54 78 7 656 6 5 65 64 63 64 7 458 456 836 734 57 34 7
3467 43 7 347 43 57 43 75 4357 45 7435 7 45 73 45 7 435 7 45 7
34 57 4 57 45 73 45 7 435 7 435 7 45 7 45 7 45 7 45 97 78 0 87
43 2 978 0 98 2346 28 9 0 8 7 65 5 4 3 4 6 7 8 76 5 4 6 8 98
76 5 43 6 78 76 54 78 7 656 6 5 65 64 63 64 7 458 456 836 734
57 34 7 3467 43 7 347 43 57 43 75 4357 45 7435 7 45 73 45 7
435 7 45 7 34 57 4 57 45 73 45 7 435 7 435 7 45 7 45 7 45 7 45
97 78 0 87 43 2 978 0 98 2346 2
```

Grading Guidelines: This lab is graded on a scale of 0-3 points, assigned as follows:

0 points: Student is absent or does not appear to have completed any work for the lab

1 point: Student has written only one program, but it does not compile or run at all due to errors.

2 points: Student has written (or attempted to write) both programs, but only one compiles and runs without error.

3 points: Student has written both programs, and they both compile and run correctly, without any apparent errors.