CS 1400 - Lab #1

Maximum Points: 10 pts.

Topics

1) Problems Solving, Arithmetic Expressions, Input/Output.
2) Getting input from user.

Coding Guideline (You will be graded on this)

1) Download the template file Lab1.java from Canvas and fill-in-the-blanks to create your Java program.
2) Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
3) Keep identifiers to a reasonably short length.
4) Use uppercase for constants. Use upper camel case for classes. Use lower camel case for all other identifiers (variables, methods, objects).
5) Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
6) Use white space to make your program more readable.
7) Use comments to explain how the parts of your program work.

Problem Description

Your friend Jenny has a class that gives three tests. She would like you to write a program that will take the three test grades as input and tell her what her average test grade is. For this Lab you are required to write a program that will read three test grades from the user and then calculate and print the average of those grades.

Sample Output

Below is an example of what your output **should look like** when this lab is completed. All text in bold represents user input. The **RED** text is your **INPUT** from the keyboard. You don't have to round the number at specific decimal point.

Test Case 1: [90, 91, 92]

```
Enter the score on the first test: 90
Enter the score on the second test: 91
Enter the score on the third test: 92
Your average score is: 91.0
```

Test Case 2: [90, 90, 92]

```
Enter the score on the first test: 90
Enter the score on the second test: 90
Enter the score on the third test: 92
Your average score is: 90.6666
```

Step 1: Getting Started with Header Documentation

For all assignments and labs in this semester, you should include header documentation in your source code files. Please put a header at the head of your file as follows, and fill in the information:

```
/*------------------------------------------------------------------------------------------
// AUTHOR: YOUR NAME
// FILENAME: TITLE OF THIS SOURCE FILE
// SPECIFICATION: DESCRIPTION OF THIS PROGRAM
// FOR: CS 1400- Lab #1
// TIME SPENT: HOW LONG IT TOOK YOU TO FINISH THIS LAB
//-----------------------------------------------------------------------------------------*/
```

Step 2: Setting Up a Scanner for Input

Since you are required to read in the three test grades from the user, you will have to use a `Scanner`. Import the `Scanner` class from the `java.util` library (see line 1 below for an example) and create a `Scanner` object (see line 2 below for an example) to get input from the keyboard (`System.in`).

```
1 import java.util.Scanner;
2 Scanner input = new Scanner(System.in);
```

Step 3: Declaring Variables

Examining the problem, we see that we will need three inputs from the user. We will need variables to hold all the inputs. For this Lab, let's assume that all the test grades will be integers. Therefore, we will need three `int` variables to hold the three test grades. Remember, if you need more than one variable of the same type, you can declare them in the same statement separated by commas. For example, if we needed two double variables, we could make:

```
double var1, var2;
```

TODO: Declare three `int` variables to hold the three test grades. Be sure to give them appropriate names like `test1`, `test2`, etc. rather than `x`, `y`, etc.

Additionally, looking at the problem, we see that we have the number 3 occurring in the problem. Rather than simply using this number in the program when needed, it is preferable to declare a constant variable to hold the number so that when it is used in the program, it will be clear what the 3 refers to. To create a constant, you must use the keyword `final` in front of the declaration. Also, it is customary to use ALL_CAPS for the name of the constant. For example, if we wanted a constant to hold the value PI, we would declare

```
final double PI = 3.14159;
```

TODO: Declare an `int` constant to hold the value 3, the number of tests. Be sure to give the constant an appropriate name like NUM_TESTS.

Finally, when looking at a problem you may need variables to hold the solution or some intermediary steps. For this problem we need to calculate an average. We will need a variable to hold the average. Usually, the average of values can contain decimal values, so you will need to declare a `double` variable to hold the average.

TODO: Declare a `double` variable for an average value

Step 4: Getting the Input

Now that we have the needed variables declared, we are ready to use the `Scanner` we created to get the input from the user. Before reading in the input though, it is important to give the user a prompt, so the user knows what they are expected to enter. Then, we use the `Scanner` object with the appropriate method to read in the value and store it in a variable. For example, to prompt and read in the first test score, we would use:

```
System.out.print("Enter the score on the first test: ");  // prompt
test1 = input.nextInt();  // read the next integer
```

, where the declared variable `test1` will hold the score for the first test and in is the `Scanner` object.

TODO: Follow the example above, write the prompt and read the input for all THREE tests.

Step 5: Calculate the Average

After reading the three input values from the user, we can use them to calculate the average. To do so, we add up all the values and divide them by the number of tests. Naively, this would be

```
average = test1 + test2 + test3 / NUM_TESTS;
```

However, due to operator precedence rules Java will do the division, test3 / NUM_TESTS, before the addition, which will give the wrong result. To force Java to do the addition first, we have to use parentheses

```
average = (test1 + test2 + test3) / NUM_TESTS;
```

This will calculate the average, but there is still a problem. Assume the test grades are 90, 90, and 92, then the average will be 90.6666, but Java will give the answer as 90 (you should run the program and print the result to verify). This is because all the variables are integers and so Java does integer division. To force Java to do decimal division, we have to cast one of the variables to a `double`. Remember to cast a value to another type you put the type you want to cast to in parentheses before the value. So, let's cast NUM_TESTS to a double.

```
average = (test1 + test2 + test3) / (double) NUM_TESTS;
```

Note: Casting sometimes is a risky operation. Please check the textbook for details.

TODO: Calculate the `average` test score in your code.

Step 6: Display Results

Now that we have calculated the result, we need to show it to the user. Use a `System.out.println` statement to display the average score to the user. Be sure to have a statement explaining what the number is. That is, don't just print the number. For example, if we wanted to print the first test score, we would use the following statement:

```
System.out.println("Your first test score: " + test1);
```

TODO: Print the result (`average` score) with a helpful message.

Step 7: Submit Your Lab1.java to Gradescope

Please submit your Lab1.java to the “Lab 1” link on Gradescope. Make sure it is compiling and producing the expecting outputs. You are done.