Assignment 1: Fundamental Java Data Types

Requirements:
- Create a Java project named yourStudentId_HW1.
- Write a single class named Tester_HW1 and adding main method in your class.
- For each question below, write your code in the main method.
- Your code must be properly formatted with sensible variable names! Refer to the text for code format examples.

1.   Write a program that does the following:
   A. Create 7 variables, all variables are primitive number types in Java
   B. Initialize each variable with any appropriate value.
   C. Print out the name of each variable and its value.
   D. Modify the value of each variable with an assignment statement and print out the names of the variables and their
        new values.

```
//For 1.D
intVar = 5;
System.out.println("intVar= " + intVar);
```

2.   Write a program that does the following:
   A. Create 3 constants, each of them is primitive number types in Java. Print the name of the constant and its value.
   B. What happens if you try to assign a value to a constant? Write your explanations in Java comment.

3.   Adding (incrementing) or subtracting (decrementing) the value one from an integer variable is a common, everyday
   operation. To increment an int variable x, we could code

   ```
   x = x + 1;
   ```

   As an alternative, we could use the special operators ++ and -- to increment and decrement a variable.
   A. Use the first method to increment x in the program below. Print the value of x after incrementing.
   B. Use the ++ operator to increment y in the program below. Print the value of y after incrementing.

4.   The output of the following program is 36.0.

```
int age1 = 18;
int age2 = 35;
int age3 = 50;
int age4 = 44;

double averageAge = (age1 + age2 + age3 + age4) / 4;
System.out.println(averageAge);
```

That is not the real average (it should be 36.75). The incorrect result is obtained because integer division is being used.
Floating-point division should be used instead.
Fix the program above so that it shows the correct result.

5.   The output of the following program is 869

```
double probability = 8.70;
int percentage = (int) (100 * probability);
System.out.println(percentage);
```

Computers represent numbers in the binary system. In the binary system, there is no exact representation for 8.70, just as there is no exact representation for 1/3 in the decimal system. The representation used by the computer is just a little less than 8.70, so 100 times that value is just a little less than 870. When a floating-point value is converted to an integer, the entire fractional part is discarded, even if it is almost 1. As a result, the integer 869 is stored in percentage.

Fix the program above so that it displays the correct result. Remember that you can use Math.round to convert a floating-point value to its *closest* integer.

Submission: Submit your project as "zip (or rar) file" via WM5. No other submissions will be graded.

Reminder: Please zip the whole project

Deadline: 2019/09/29 (for both Mon56 and Tue23)