Assignment 2 – Working with Variables

Problem 1 – A simple greeting

This problem will give you a chance to get your hands dirty with a ConsoleProgram. Write a program which asks the user for his or her name. The program should reply with a greeting to that user. In addition, if the user has your own name, you should reply with a special greeting. A sample run of the program might look like this, where the bold text has been entered by the user.

> Hello! What is your name? **Thor**
> Hi Thor, it's good to meet you.

Another run might appear

> Hello! What is your name? **David**
> Weird, that's my name too.

You do not need to copy the wording exactly. Feel free to write your own greetings and responses, as long as it meets the requirements in the above paragraph.

Problem 2 – Computing mean and variance

(a) Write a program which computes the average of a list of numbers input by the user. The average, or **mean** of a list of numbers is the sum of the numbers divided by the number of numbers. You should ask the user to input numbers in sequence until they enter -1, at which point you should report the average of all the numbers entered prior to the -1. A sample run of the program might appear:

> Enter a number (or enter -1 to finish): **4**
> Enter a number (or enter -1 to finish): **4**
> Enter a number (or enter -1 to finish): **5**
> Enter a number (or enter -1 to finish): **6**
> Enter a number (or enter -1 to finish): **-1**
> The average is 4.75.

(b) The **variance** of a list of numbers is a measure of how far those numbers are from their mean. A high variance indicates that the numbers are widely spread out, while a low variance indicates that the numbers are all very close to each other. In particular, a variance of 0 occurs exactly when all of the numbers are exactly the same, The variance is never negative.

For a list of numbers $x_1, x_2, \ldots, x_n$, the variance may be computed by the formula

$$\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{n} - \left(\frac{x_1 + x_2 + \cdots + x_n}{n}\right)^2$$

For example, the variance of the numbers $10, 11, 12$ is computed as

$$\frac{10^2 + 11^2 + 12^2}{3} - \left(\frac{10 + 11 + 12}{3}\right)^2$$

which turns out to be $2/3$.

Another way you may find helpful to think of this computation: the first term is the average of the squares of the inputs. The second term is the square of the average of the inputs.

You should modify the above program program to also print out the variance of all the numbers entered before the -1. A sample run might appear the same as above, with the additional line at the end,

> The variance is 0.6875.

As a side note, the square root of the variance is known as **standard deviation**, and is commonly used when discussing statistics (such as test scores) to indicate how widely spread the data is. If you are interested in also computing standard deviation, you should look into the method `Math.sqrt()`, which takes a single double as its argument and returns a double.

Challenge Problem – Reverse guessing game

In class, we wrote a program which challenged the user to try to guess its hidden number, and at each step the program told the user whether the guess was high or low. In this problem, we write a program to play the same game, but from the other side. The user thinks of a random number, and the program makes guesses. After each guess, the user tells the program whether the guess was high, low, or correct. A sample run might appear as follows:

> Think of a number between 1 and 100. I guess 50. Is that high, low, or correct? **high**
> I guess 25. Is that high, low, or correct? **high**
> I guess 12. Is that high, low, or correct? **low**
> I guess 18. Is that high, low, or correct? **high**
> I guess 15. Is that high, low, or correct? **correct**
> Good game!

The program's strategy is to keep track of a range in which it knows the number is in, and to always guess at the middle of that range. For example, in the above run it followed the following reasoning:

- The initial range is 1 to 100. The middle of this range is $(1 + 100)/2$, which I'll round down to 50.
- 50 was too high. That means I now know the number is in the range 1 to 49. The middle of this range is $(1 + 49)/2$, so I'll guess 25.
- 25 was too high. That means I now know the number is in the range 1 to 24. The middle of this range is $(1 + 24)/2$, which I'll round down to 12.
- 12 was too low. This means I now know the number is in the range 13 to 24. The middle of this range is $(13 + 24)/2$, which I'll round down to 18.
- 18 was too high. So the number must be in the range 13 to 18. The middle of this range is $(13 + 18)/2$, which I'll round down to 15.

It's a very good accomplishment to get a basic version of the program running. If you can do that, you can try adding the following features:

- If the user enters something other than "high", "low", or "correct", prompt them again until they enter one of the acceptable values.

- If the user has given input which cannot describe any actual number (for example, the user never says that any guess is correct, leading to a range from 25 to 24), then the program should let the user know that they are a dirty, dirty cheater.