

Program #2
Introduction to OOP with Java
COP 2551 – Summer 2011
Strings and Control Structures

Objectives:

- Gain more experience with interactive programming (prompts and replies)
- Gaining experience in writing a Java program from scratch
- To gain experience with Strings and Control Structures
- To gain experience with validation of inputs
- To gain experience with loops and conditional expressions.

Functionality:

Design and implement a solution to the following specifications:

Task 1: Prompt the user for his/her name (first and last). Be sure to get these in two separate prompts. Then,

Task 2: Now, you are to first prompt the user for his/her classification as an integer: (1 = Freshman, 2 = Sophomore, ..., 5 Graduate Student). From the input, determine the classification as a String. This task is to be accommodated in a loop. You are to give the client (me) three chances to input a valid number. If I fail to input a 1, 2, 3, 4, or 5 on the third try, you are to gracefully terminate the program and provide the message, "Sorry, Charlie. Get a life!"). Recognize that I may give a correct number on the first, second, or third try, however. If so, your program should proceed as follows.

Task 3: You are to then prompt the user to input three numbers (integers) three times but each one entails different processing.

You are to create the average of the three numbers. You are to create a method in public class Main called **average** to compute the average, which will return to you a letter grade of the average computed within this method. You are to specify in your prompt whether you want these three integers submitted one at a time or if you want to read all three in via one input line with integers separated by spaces. Your choice, but be clear in your prompt.

In **average**, you are to determine the letter grade (and return it) using the criteria:

- 90 through 100 is A
- 80 through 89.9 is a B
- 70 through 79.9 is a C
- 60 through 69.9 is a D
- Below 60 is an F

For the first set of numbers, you are to determine the letter grade above using an if...else if construct.

For the second set of three numbers that you read in, you are to determine the letter grade using a nested-if.

For the third set of three numbers that you read in, you are to determine the letter grade by using a Switch statement.

Task 4: After this, you are to display the results.

Your outputs from each one will be nearly identical.

Your output should have the following format:

<skip two lines> (this means two blank lines are to appear in the output)
<name> <classification> earned a <print the letter grade> computed using an if...else statement.

<name> <classification> earned a <print the letter grade> computed using a nested-if construct

And

<name> <classification> earned a <print the letter grade> computed using a Switch construct.

Skip two lines after displaying these three lines of output.

Task 6: When the program is to be terminated, simply issue the text: "That's all folks!" (be sure to include the apostrophe in That's)

As will be explained in class, you will need to supply comments at the beginning of your source code using **Javadoc** and also internal documentation where appropriate. Be certain to tag (provide a comment for) closing braces on methods and classes (scope terminators). You must also turn in your UML (see textbook).

Turn your zipped assignment in Blackboard as you have done like, project2jschmuck

As usual, this programming assignment will be accepted after the due date but with heavy penalty. If you have any questions, **DO NOT WAIT UNTIL IT IS TOO LATE!** This is a fun assignment. Enjoy the learning process! ☺

Programming Guidelines:

- Your programming style should adhere to the design guidelines in the Appendices of your textbook and the recommendations made in class.
- Be sure you add a comment to the close brace of each method and each class, as mentioned in class (scope terminators)

- **Javadoc** – will be briefly discussed in class. Be careful. There are several ways to produce it. I want a very specific approach to be used that will generate the .html files that I'm looking for. See my web page for the procedure. It is easy.

- In your source program, be sure to include your name in the information in your program header information

- Be sure to have a consistent indentation approach to your source code

- **UML:** See book. For this program, it will be a simple box with three compartments: one for the class name (Main) and one for naming the attributes and one for naming the methods. You will have two methods: main and average. See examples in book. This attempt is only to get you started. You **must use** power point or Word. Save the file as a .pptx or .docx and drag it into the project folder, project2yourname PRIOR to zipping it all to me.

- **Pseudo-code.** Essential for non-trivial programming efforts. Do your pseudo-code in Word. Include the .docx file in your project folder before zipping it to me.