CS 1400 - Assignment #3

Maximum Points: 20 pts

Topics:

- Using Classes and Objects (The String class)
- Using conditional (if, if-else, ii-else-if) statements
- Writing loops

Coding Guideline (You will be graded on this)

1) Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
2) Keep identifiers to a reasonably short length.
3) Use uppercase for constants. Use upper camel case for classes. Use lower camel case for all other identifiers (variables, methods, objects).
4) 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.
5) Use white space to make your program more readable.
6) Use comments to explain how the parts of your program work.

Important Note: All submitted assignments must begin with a descriptive block comment (multi-line comments) similar to the one shown below. It must contain your name and the other information illustrated. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment.

```
/*-------------------------------------------------------------------------------------------
// AUTHOR: YOUR NAME
// FILENAME: TITLE OF THIS SOURCE FILE
// SPECIFICATION: DESCRIPTION OF THIS PROGRAM
// FOR: CS 1400 - ASSIGNMENT #3
// TIME SPENT: HOW LONG IT TOOK YOU TO FINISH THIS ASSIGNMENT
//*/
```

Part #1: Written Exercises: (5 pts)

Note: The answers to the following questions should be typed in the block of comments in the Assignemnt3.java file, right after your header. Please make sure they're commented out (green). Type them neatly and make them easy to read for the graders.

1. Rewrite the following set of if statements using a nested if (NOT an if-else-if) structure. **Requirements:** You must use only the > (greater than) relational operator to create your `boolean` expressions and your `else` statement(s) must include a single Java statement (one command) not a block of them. Assume that **all** scores are integers.

```
if (score >= 90) grade = 'A';
if (score >= 80 && score < 90) grade = 'B';
if (score >= 70 && score < 80) grade = 'C';
if (score >= 60 && score < 70) grade = 'D';
if (score < 60) grade = 'F';
```

2. Rewrite the following if-else-if statement as an equivalent **switch s**tatement. Requirements: your must have only 4 `case` statements in your program, not more.

```
if (letter == 'A' || letter == 'a' || letter == 'B' || letter == 'b')
       System.out.println("You are doing good!");
else if (letter == 'C' || letter == 'c' || letter == 'D' || letter == 'D')
       System.out.println("You need to study more!");
else
       System.out.println("Maybe next time!");
```

Part #2: Programming: (15 points)

Write a program called Assignment3 (saved in a file **Assignment3 .java**) that asks a user to enter two strings.

First, the program prompts:

Please enter a string.

The program should read the string, and prompts:

Please enter another string.

The program reads both strings and prints a menu showing the following command options:

Command Options

Option a: checks if the length of the two strings are the same.
Option b: checks if two strings are the same or different.
Option c: checks which string is lexically larger, or if they are the same
Option d: prints out the first character (index 0) of each string
Option e: prints out a new string consisting of the first string concatenated (appended) with the second string.
Option f: prints out two strings using upper case letters. Option
q: quit the program.

After that, the program asks the user to enter one of the menu options prompting:

Please, enter an option from the menu.

Here, the program executes the chosen tasks and outputs the results. In case the user wants to continue, the program asks the user to enter one of the menu options again., otherwise it prints “Goodbye!”.

Here is the sample output your program should produce when the user enters the strings shown in bold:

Sample Output:

Please enter a string.
apple
Please enter another string.
orange

Command Options
---
a: find if the lengths of the strings are equal
b: find if the two strings are the same

c: find which string is lexically larger
d: print the first character of each string
e: concatenate the two strings
f: print both strings in uppercase
q: quit this program

Please, enter an option from the menu.
a

The lengths are not the same.

Please, enter an option from the menu.
b

The two strings are different.

Please, enter an option from the menu.
c

The second string is lexically larger.

Please, enter an option from the menu.
d

The first character of the first string is a
The first character of the second string is o

Please, enter an option from the menu.
e

The concatenation of two strings is "appleorange" Please,

Please, enter an option from the menu.
f

The first string using upper case letters: APPLE
The second string using upper case letters: ORANGE

Please, enter an option from the menu.
x

Invalid option.

Please, enter an option from the menu.
b

The two strings are different.

Please, enter an option from the menu.
q

Goodbye!

The other potential answers are:

a. The lengths are the same.
b. The two strings are the same.
c. The second string is lexically larger OR The two strings are the same.

Requirements: MAKE SURE you are using a switch statement for the menu. In addition, your program should keep asking for input until user chooses to quit.

Helpful hints for doing this assignment:

- work on it in steps
- always make sure your code compiles before you add more code
- remember that options can be called in any order

Submit your homework by following the instructions below:

****

- Submit your **Assignment3.java** file on GradeScope. Your assignment will be graded only if it is submitted there, NOT on Blackboard.
- Assignment3.java should have the following, in order:
  - In comments, the assignment Header described in "Important Note".
  - In comments, the answers to questions 1-2 in Part #1.
  - The working Java code requested in Part #2.
- The Assignment3.java file must compile and run as you submit it. You can confirm this by viewing your submission results.

Important Note: You may resubmit as many times as you like until the deadline. Only your last submission will be considered.

NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!