Project 4: Stress Quiz

Due: Monday, Feb. 17th at 11:59 PM

Description: Experts believe that American college students may suffer from "greater levels of stress and psychopathology than any time in the nation's history."[1] To bring a bit of attention to this issue, we will implement an automated version of a quiz from Psychology Today: http://www.psychologytoday.com/blog/just-listen/201010/are-you-stressed-out-take-the-quiz.

The purpose of this quiz is to help you gauge your level of stress. The quiz is a bit long for our purposes, so we will cut it down from twelve to three questions and adjust the scoring proportionately.

Objectives: Your program will be graded according to the rubric below. Please review each objective before submitting your work so you don't lose points.

1. Create a new project and class in Eclipse and add the main method. (5 points)
2. Construct a Scanner to read input from the keyboard. (5 points)
3. Prompt the user with three questions from the Psychology Today quiz, and use the Scanner to read their responses. (15 points)
4. Use conditional statements, logical operators, and String methods to check if each user response is either "yes" or "y" & "no" or "n" with arbitrary capitalization. (30 points).
5. Use an accumulator to track the number of statements that the user agrees with. (15 points)
6. Use a cascading if-else statement to print the result of the quiz that corresponds to the value stored in the accumulator. (20 points)
7. Use meaningful variable names, consistent indentation, and whitespace (blank lines and spaces) to make your code readable. Add comments where appropriate to explain the overall steps of your program. (10 points)

[1] Source: https://www.psychologytoday.com/us/blog/theory-knowledge/201402/the-college-student-mental-health-crisis. If you think you may be suffering from excessive stress, anxiety, depression, or any other mental illness, the University of Oklahoma has excellent resources that can help: https://www.ou.edu/ucc.

Sample Output: Below is an example run of the program. The user input is in bold.

```
Psychology Today Quiz: Are You Stressed Out?
------------------------------------------
Do you agree with the following statements?

1. I am losing my sense of humor.
Yes
2. I find it more and more difficult to see people socially.
no
3. I feel tired most of the time.
y

You are possibly stressed out.
```

Choose any three of the twelve statements from the Psychology Today quiz to include in your program. The final line of output should depend on the number of yes responses. The possible final lines are shown in the following table:

Yes responses	Final output line
0	You may be more exhausted than stressed out.
1	You may be beginning to stress out.
2	You are possibly stressed out.
3	You are probably stressed out.

Note that you can print a blank line by calling System.out.println with no argument.

Getting Started: If you need help creating a new project or class in Eclipse, see the Project 1 instructions. If you need help naming the class or writing the main method, see the Project 2 instructions. If you need help constructing a Scanner object or calling Scanner methods to read user input, see the Project 3 instructions.

Checking User Responses: Allow the user to agree with each statement by inputting “yes” or “y” with any of the letters upper or lower case (e.g., “yEs” counts as agreeing). If the user enters anything else, assume they disagree with the statement. Use logical operators to check for different inputs. If you use the correct comparison method in the String class (see the next section), you will only need to check for a few responses.

String Comparisons: In class we talked briefly about comparing string variables. If you recall, string variables must be compared differently than variables that store primitive data. If we have two integer variables int1 and int2, for example, we can check if they store the same value using the equality operator:

```
int1 == int2
```

Comparing string variables like this is not illegal, but it won’t compare the content of the strings themselves. Rather, it compares the addresses where the strings are stored in memory. To compare the content of the strings, we need to use methods of the String class. If we have two string variables str1 and str2, for example, we can check if the String objects to which they refer have the same contents with an expression such as

```
str1.equals(str2)
```

Note that the String class has more than one method that can be used to compare String objects. The equals method is not a good choice for this program because the user can enter "yes", "Yes", "yEs", "yeS", "YEs", "YeS", "yES", and "YES", and all of these inputs should be considered agreeing with a statement. Review the String class API documentation for a more convenient method to use in your program.

Use an Accumulator: In the last project, you used multiple variables to perform a computation. For this project, you only need a single variable to store the number of yes responses. If you initialize this variable to 0, you can simply increment it (i.e., add 1 to it) every time the user enters "yes."

A variable that is used like this to keep a running sum is called an "accumulator." Using an accumulator instead of multiple variables can save memory and reduce code complexity. Consider, for example, the code below, which uses an accumulator to calculate the total cost of some items at the grocery store.

```
double totalCost = 0;          // Initialize the accumulator
totalCost = totalCost + 6.40;  // Add price of coffee
totalCost = totalCost + 5.89;  // Add price of Cooler Ranch Doritos
totalCost = totalCost + 8.99;  // Add price of cat food
```

After this code is executed, the variable totalCost will contain the sum of the prices of the items. Space for only a single variable is needed in memory, and the logic of the code is easy to understand.

Implementation Suggestion: The code you write for each quiz question will implement the same steps: prompt the user, read their response, and increment the accumulator if necessary. Write and test the code for a single question first. After you are satisfied that it works correctly, copy and paste the code and make the necessary modifications for the other two questions.

Upload Code to Zybooks: After you complete each project, you need to upload your source code to Zybooks so it can be graded. If you created the folder structure described earlier, your code is in the folder "Intro to Programming\Projects\Project 4\src". The code is contained in the file Project_4.java. Note that .java files are simply text files that contain Java code. They can be opened with any text editor (e.g., Notepad or TextEdit).

Upload your .java file to Zybooks on the Project 4 assignment page. This requires a few steps. Click the "Submit Assignment" button in the top-right corner. This will reveal a button near the bottom of the page with the text "Choose File." Click this button and browse to the location of your .java file. Finally, click the "Submit for grading" button below the "Choose on hard drive" button.

If Zybooks won't accept the file, make sure you're submitting a .java file, not a .class file. Make sure to see if the output you obtained and the output which I have mentioned match to obtain complete grade.

Submission Link:

https://learn.zybooks.com/zybook/OUCS1324Winter2020/chapter/21/section/4