Project 2

CS 1323, Fall 2017

# Learning Objectives

1. Use an if or if-else statement in a program. (10 points)
2. Use a nested if-else statement in a program. (20 points)
3. Keep a running sum (accumulator) in a program. (10 points)
4. Get String input from the user (10 points)
5. Perform a String comparison using .equals or .equalsIgnoreCase (10 points)
6. Show output to the user (10 points)
7. Create a program that administers an online quiz correctly. (20 points)

10 points will be awarded for the documentation of your program. That means using good names for variables, comments, proper and consistent indentation of code, and meaningful use of whitespace.

When your program is completed and running, upload the program in the dropbox on Canvas

Due: Friday, September 15 by 11:59 p.m.

# Description

Psychology Today has an interesting quiz to determine if you are stressed out: <http://www.psychologytoday.com/blog/just-listen/201010/are-you-stressed-out-take-the-quiz>

This quiz can be automated, so you can help your friends tell if they are stressed out or not.

The quiz is pretty long (12 questions), so you may cut it down to as little as three questions and adjust the scoring proportionately. I’ll let you pick the three questions you think are the best indicators of stress.

For a three question quiz, the scoring would be:

0 points: More exhausted than stressed out

1 point: Beginning to stress out

2 points: Possibly stressed out

3 points: Probably stressed out

A sample user input is shown below:

Answer yes or no to the following questions

I am losing my sense of humor.

**Yes**

I find it more and more difficult to see people socially.

**no**

I feel tired most of the time.

**yes**

You're possibly stressed out

# Coding Notes

I see a lot of code from beginning programmers that looks like the code below (I’m using generic variable names so I don’t write your program for you).

Scanner input = new Scanner(System.in);

int data1 = input.nextInt();

int data2 = input.nextInt();

int data3 = input.nextInt();

int data4 = input.nextInt();

int sum = data1 + data2 + data3 + data4;

There’s nothing wrong with that code in terms of functionality, but it is creating four variables when only one is needed. Here is an alternate way to write the code using one variable repeatedly and a running sum. This technique is very common in programs and demonstrates a better understanding of how memory works.

Scanner input = new Scanner(System.in);

int sum = 0; // this is a running sum

int data = input.nextInt();

sum = sum + data;

data = input.nextInt(); // location data is being reused

// this is OK because we were through with the old value in data

sum = sum + data;

data = input.nextInt(); // location data is being reused again

sum = sum + data;

data = input.nextInt(); // location data is being reused again

sum = sum + data;

At first you might thing this code is worse than the code above because it is longer and does the same thing. However, it uses less space in memory (2 variables instead of 5). This means it has less memory load (i.e. fewer details that the programmer needs to remember). The first code is less complicated, however it is not the way that any proficient programmer would write the code. So I want you to try the second method of writing code in this project.

# Implementation Suggestions

Write one question first, get the user answer, and save it to the accumulator. Once you have that working perfectly, you can copy and paste for the other questions. You’ll have to make some minor modifications to the code (make sure you test after you do this—copy and paste tends to get people into trouble), but it should work easily.