Laboratory 4

CS 1323, Spring 2015

# 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. (10 points)
3. Use a while loop in a program (20 points)
4. Keep a running sum in a program. (10 points)
5. Use String input and proper comparisons (.equals, .equalsIgnoreCase, or == ) (10 points)
6. Create a program that administers an online quiz. (30 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.

Section 10: When your program is completed and running, upload the program in the dropbox on Janux. Then have the teaching assistants check it to get credit for the lab.

Section 1: Upload your program to the dropbox on Janux.

Due Date: February 11, 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

The whole quiz should be able to be repeated. The program should keep track of how many people who took the quiz fell into each of the four categories and report these numbers to the user when the program is stopped.

# 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.

Scanner input = new Scanner(System.in);

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

int data = input.nextInt();

sum = sum + data;

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

sum = sum + data;

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

sum = sum + data;

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

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), and is therefore less complicated. It is also the way that any proficient programmer would write the code. So I want you to try this 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 cut and paste for the other three questions. You’ll have to make some minor modifications to the code, but it should work easily. Once you have that running, then you can make the survey repeat multiple times, and count the number of your friends in each category.