**CS341 Software Engineering**

**Homework 2**

**Write your code for Part II in Java. Submit your code and output screens to Canvas.**

**Part I: Complete the reading assignments (chapters 1-3) from the following url.**

<https://git-scm.com/book/en/v2>

Read Chapter 1: Getting Started of the ProGit book.

Read Chapter 2: Git Basics of the ProGit book.

Read Chapter 3: Git Branching of the ProGit book.

**Part II: Debugging Java Programs with Eclipse and Creating Javadoc comments**

In this section of the homework assignment, you will create two GUI apps. You should design the visual elements of both apps anyway you wish. Embed your code with well written **Javadoc** comments and generate a **Javadoc** for both applications. In addition, develop both these apps using **WindowBuilder** and Eclipse's **Debug** perspective. When using the Debug tools in Eclipse, practice the following:

* Suspending, Resuming and Terminating the program
* Setting Breakpoints
* Step Into, Step Over and Step Return
* Viewing Variable Values
* Evaluating Expressions

App 1: Use WindowBuilder to create the following Scrabble App.

* Allow the user to enter the letters of 4 scrabble tiles.
* Display **all** the ways to assemble these tiles. Each “word” should contain all the tiles.
* Detect and display an error when the user tries to enter more than 7 scrabble tiles.
* Detect and display an error when the user tries to enter non-letters of the alphabet.
* Fully test your app.

App 2: Use WindowBuilder to create the following Password strength app.

Passwords are often characterized as weak or strong based on the length of the largest “block in the sequence of characters. Design your app so that the user is allowed to enter a password of no less than 8 characters and no more than 12 characters. Detect and display an error when the user tries to enter a space or a password of invalid size.

Compute the length of the largest "block" in the password. A block is a run of adjacent characters that are exactly (case sensitive) the same.

Display the length of the block. If the length of the largest block is 2 or less, recognize this as a decent password. Otherwise, suggest to the user that they should shrink the block.

Examples

Password: **7hoopla** Output: The largest block in the password is 2. This is a decent password.

Password: **xyyyyyyy2** Output: The largest block in the password is 7. This password can be made stronger by reducing this block by 5.