**Project 1: Introduction to C++ Programming**

(Due September 12 at 4:45pm)

Learning Objectives:

* Write C++ programs in a modern software development using the CLion IDE and version control with Github repositories.
* Perform basic computational operations such as i/o,
* Apply control structures such as if statements, loops, and error messages.
* Implement and manipulate the contents of a 2D vector.

For Project 1, you will complete a C++ program in CLion and use a Github repository where you will “push” your C++ program (see Appendix).

**Battleship**

Write a program named “Battleship.cpp” that will print the matrix 3 x 3 from the user and repeatedly ask the user to guess a location on the matrix. Using a random number generator, you will choose two numbers between 1 and 3, inclusive, and mark that location on your 3 x 3 matrix but not show this to the user. Each time the user guesses a location, you will update the location on your matrix, print the matrix, and keep track of how many guesses the user has made. If the user correctly guesses the location, the program will print the updated matrix, output the total number of guesses, and end the program.

There will be no check for whether the location guessed by the user has already been chosen (unless you want to make it part of your program, which I encourage only if you complete the rest of the project).

You are required to use vectors.

One example output for 3x3 matrix might be:

! !

~~~~~

! !

~~~~~

! !

Location (row[1-3] column[1-3]): 2 2

! !

~~~~~

! X !

~~~~~

! !

Location (row[1-3] column[1-3]): 1 3

! ! O

~~~~~

! X !

~~~~~

! !

You sunk my battleship! (2 guesses)

Project 1 Invitation Link: <https://classroom.github.com/a/hPL545OP>

**Implementation Notes:**

Any program that does not compile without errors (warnings okay) will not be graded and be given an automatic 0 points for that part.

**Comments and Style:**Although there will be no formal policy on commenting and style, the reader should be able to easily follow the main purpose of the code. Each set of code that does something significant must be commented. The variable names should be easily recognizable and acronyms should be avoided if possible.  
  
*Do not be surprised if help is not forthcoming if your code is poorly commented and/or difficult to follow. You have been warned.*

**Project Submission:**

You will develop your program for this project with CLion connected to a remote Github repository given to you. No credit will be given for late submissions.

**Pledged Work Policy:**

Assignments in Computer Science courses may be specified as "pledged work" assignments by the professor of the course. When an assignment is specified as "pledged work" the only aid that the student may seek is from either the course professor or TAs (including CS Center tutors) that the professor has explicitly specified. On "pledged work" assignments the student may not use the services of a tutor.

For this project, you and your partner will develop code together into shared repositories, so you will see and share work. In addition, you may discuss only basic C++ and Java syntax and general computer science concepts with everyone else. Any other communications of the project (e.g., giving your code to someone else or seeing someone else’s code) are strictly prohibited except with the professor and TAs of the course. Your code and your implementation of the project must be the product of your own work and that of your partner.

**Appendix:**

1. Click on the “Clone or download” button and a small panel will open. Copy the url that looks like: <https://github.com/WFCSC112AlqahtaniFall2019/project1-username.git>

Do not close the browser.

1. Open CLion. From the “Welcome to CLion” window, choose “Check out from Version Control” | Git. Enter the url from the previous step.
2. When asked whether you would like to open a directory, click the “Yes” button.
3. File | New Project…
4. In Location, use the explorer button to move to the newly created directory (For example: project-1-part-1-jobs-steve-musk-elon). In Language Standard, choose C++11. Click the “Create” button.
5. There will be window stating that the directory is not empty. Click the “No” button. Then there will be a window asking about how you would like to open the project. Choose “This Window”.
6. At this point, a “Hello, World!” C++ program should pop up.
7. VCS | Git | Add.
8. VCS | Git | Commit File…
9. Enter a Commit Message (most people’s first commit message is “Initial commit.”) then click the “Commit” button.
10. There will be a window that asks about adding an .xml file. Click the “No” button.
11. VCS | Git | Push...
12. Click the “Push” button.
13. Go back to your browser from Step 5 and refresh it. You should now see a main.cpp in your repository. You can click on it and see your code. If you got to this step successfully, that means that you are able to deposit your code to your Github repository.
14. Continue writing your program, periodically repeating steps 13, 14, 16, and 17.