

# OOP Assignment #1

Samuel Jonathan Orman-Chan (25659005)

March 28, 2022

## 1 The Code Itself & Git

To begin, my first task was to retrieve the base code from Blackboard and unzip it. This was accomplished fairly easily and with minimal fuss. From there it was a task of creating the git repository and connecting that to GitHub. This was completed with `git init & git remote add origin https://github.com/SJO-C/CMP1903M.git`. This then required the use of OAuth to authenticate to my GitHub account which was completed in a straightforward manner. Subsequently, I was able to commit and push to the online repo that I set up. Since then, I made 56 commits, some descriptive, others filled with frustration.

With regard to the code itself, error handling was a bit of a battle as I struggled to understand scope with regard to `Try, Catch & Finally` thus resulting in an awful lot of code that could be considered redundant. Regarding comments, I commented processes as opposed to every line. This meant that some areas were very densely commented, whilst others had only a comment above the section where the code was. Additionally, I added a small amount of encapsulation for the letter counter. This was as to meet the points on the checklist. I did deviate from the brief with regard to the user interface as I felt it seemed a little cumbersome [in the brief]. Furthermore, my sentence input uses the newline character to indicate completion for analysis as opposed to an asterisk, but the asterisk indicates to the program to cease processing anything after it.

## 2 Reviews

My approach to code reviews was to download a zip file of the person's sourcecode from their GitHub Repository, and then read the code before running it on my own machine.

I would start by examining the code for any glaring logic errors before attempting to run the code. Then I would assess the user interface at launch and the subsequent menus that I faced. On the code side, I would go through program source file by program source file and check for the amount of comments and how effective I felt they were for describing functionality. Then I would look at the amount of loops versus methods like "regex" to try to analyse efficiency. Finally, I would write my report onto the person's GitHub as an "Issue" on the repository in question.

My reports would be structured as a heading for each file analysed and then some bullet points on my thoughts about the said file. An example of this can be seen at <https://github.com/LWhitehead15/ObjectOrientedProgramming-Asseessment-1/issues/5>, <https://github.com/KaiserschlachtCat/CMP1903M/issues/3> & <https://github.com/Just-A-Tea-Drinker/OOP-Assement/issues/1>.

In return, I recieved 4 reviews for my code, which I was able to act on. These actions are noted in comments next to lines which were modified based on the feedback from said code reviews. All of these can be accessed from <https://github.com/SJO-C/CMP1903M/issues>.

END