**Coding Contribution guide**

**Submitting work to GitHub**

Newly edited code must be put on a testing branch of the GitHub. This is to ensure the new contribution is compatible with the rest of the code before it can be committed to the main branch. Helping to prevent newly submitted code from causing errors in the program which did not exist before, while reducing the risk of working sections of the code being overwritten in the commit and potentially resulting in a setback in the project schedule.

Ensuring that the part of the code updated or changed is shown before submission is required. This is to ensure others editing the code are aware of the changes that have been made or the new code present. If this is not done it could result in the duplication of functionality as multiple people work on the same requirement.

The date and version of the project in which the changes were made, should also be included in the commit, to help identify where errors occur if the new code causes errors. As well a short summery of what the new addition does and what it addresses.

**Testing before submission**

Testing on the newly written code must be done to ensure it runs as desired. This will need to be done on the latest version of the program, so it is important to initiate a pull before tests are completed. The tests carried out need to ensure all main functionality of the program still works and that there has not been a considerable reduction in performance in the program due to inefficient code. For non – complete code submissions, highlight that the work is in progress and not meant to be functional.

**Meeting requirements**

When contributing to the program it is important that the code submitted meets the requirement of the assessment specification in terms of the implementation. This is ensuring that where possible: the STL library should be used, the inclusion of error and exception handling, using private member functions for common functionality and appropriate test cases and testing library is used. These requirements do not need to be in every submission of code. However, if the functionality is important and it is possible to include these requirements, they should be included to ensure we are meeting them in the specification from the start, rather than trying to fit the requirements in at the end.

**Documentation and presentation**

Before a contribution is made, it is important that it is documented effectively and with Doxygen comments where possible. This is to ensure that the newly edited code is easy to understand and read for others, while being neatly documented within the Doxygen files. Creation of class templates and separate functionality will also help with readability so must be done with all classes before submission. Classes With similar functionality need to be grouped in these files to help keep the program organised and easy to understand.

Comment length should be kept concise where possible, while ensuring the code is explained effectively and comments are spelt correctly and neatly. For example, having a one-line comment stretch too far to the right should be avoided to help readability and multiple line comments used for these more in-depth explanations of code.

Sources used -- <https://mozillascience.github.io/codeReview/contrib.html>