Text

Description automatically generated

**CS3318 Assignment 3**

**Semester 1, 2022/23**

**Student Name: Mark Kilgannon**

**Student Number: 120398166**

**Module Code: CS3318**

**Programme and Year of study: Computer Science III**

(e.g., Arts III, Food Science II)

**Wordcount: 440**

**Student Declaration**

This is to certify that the work I am submitting is my own and has been done by me solely and not in consultation with anyone else. Neither I nor anyone else have submitted this work for assessment, either at University College Cork or elsewhere. I have read and understood the regulations of University College Cork concerning plagiarism. Where breaches of the declaration are detected, these will be reviewed under UCC student conduct and plagiarism policies. Any breach of the examination rules is a serious issue and can incur penalties.

<https://github.com/Markkilgannon/Colour.git> **- I had difficulty trying to download the git log file, so I attached my GitHub link to allow the commits to be viewed**

**Introduction**

**Test Driven Development (TDD)** is software development approach in which test cases are developed to specify and validate what the code will do. TDD in simple terms is when we write the test cases first to check their functionality, if the test case fails, we proceed with writing new code to make the test case pass the test. TDD methodology can make code simple and bug free.

**The Process Of TDD**

Test Driven Development is broken in to three steps. It starts with designing and developing small tests for all the different functionalities in the code. These tests must be precise, and they must compile so the program can execute. We expect these tests to fail.

We then developed more code to ensure these tests pass. Minimal changes are usually required so the programs run.

The last step is to refactor the code. Once all the test pass successfully, the programmer will start making changes to the code to enhance the overall performance. These new additions include comments and ensuring the concepts of clean code development can be seen in the program.

**My example of my TDD process**

In this assignment, we developed a colour class that represented the colours Red Green and blue using the TDD approach. I used Unit testing to implement this approach.

The first test was zeroTestRed(). The idea of this test was to see if getRed() could handle the inputs of the specified colour component, red. The values were 0.0. This test failed at first. After writing code into the colour class, this test passed, and we could clearly see getRed() could handle these values. This same format of testing was used for the other two colours… zeroTestGreen() and zerotestBlue().

**Did you think the TDD approach benefitted the development process?**

I was in favour of the Test-Driven Development process for this assignment for two main reasons.

Adopting the TDD process helped me (the developer) become more efficient and productive. I created more tests than I usually would have which overall benefitted the running of the code. Writing the tests at the start made the code much easier in the later stages of development.

In step we are required to refactor the code. Refactoring the code as whole instead of improving it as I go along made the program more maintainable and easier to understand. I was more closely focused on improving the performance of the code than I would have been if I coded this assignment the classic way.