Comp 47480 Learning Journal – Assignment 3

1. Reflection on my Learning on the Test-Driven Development.

Traditionally software was developed by implementing the methods and then testing each method. In comparison Test Driven Development (TDD) attempts to implement and specify the tests before implementation. This is achieved by iterating through a repetitive cycle of software development. This process makes it necessary to write a failing test case before implementing any methods. As a next step we then implement just enough code to make the test case pass and then refactor before repeating the cycle. The benefits to using this type of development in that the code written when using this method of development in often clearer and cleaner due to the fact we are only writing code that it necessary. The industry standard for the implementation of unit testing is Junit4. TDD is a fundamental concept of the agile methodology

For part 3 of the practical I used the EclEmma tool. This determines the code coverage of the test cases. When I ran the code coverage the first time I got 85% which means I didn't completely follow the TDD process although most of my code was covered the invalid triangle was the branches that was causing me issues. I realised after that my logic was slightly wrong as I had implemented the code, so it only failed when all the sides were 0 instead of if two sides are added and are less than the longest side as in this instance the sides would not connect. Once I fixed this the test passed. I tried to delete the 1 on the test cases and it resulted in the coverage dropping dramatically. So, I can say that there are not any redundant test cases within my code. The problems that can occur when redundant code is added to the test cases is that bug can be introduced as when we apply test cases the code we are testing may be covered by other methods this causes regression. In this case it is important to note that code coverage could be at 100% but it may never fully prevent bugs due to regression. In terms of coverage we needed to implement statement coverage where all statements are covered at least once Also due to the method containing if Else statements we needed to ensure the branch coverage was also covered. This is where my test had passed but on analysis of all the branches had not been covered this type of coverage is normally highlighted yellow in Eclipse, so it is easier to detect.

As a technique of developing software, I can see how using TDD can benefit the design of a project. However, I do think that it's more useful to utilize TDD in larger designs rather than in this assignment as it is quite simplistic. Nevertheless, it has been beneficial to observe the TDD process.