**Reflective Report**

1. Aspect of testing that was most effective.

Ans: The greater part of the classes we run over have dependencies, and in many cases, methods assign functionality to different methods in different classes. At the point, while unit testing, these methods, on the off chance that we just use JUnit, the tests will rely upon those methods. It is required that the unit tests are independent of every other dependencies. Thus, the testing was carried out by mocking out the reliance utilizing Mockito, which was really effective. The appreciable thing about mocking is that it is verifiable that specific methods are called on the mock objects while execution of the test.

The arrange, act, assert pattern is really beneficial and effective in the unit test methods as it distinguishes setup, steps for verifying and results from what is being tested and make the code really readable. Furthermore, it implements a specific level of order whilst composing the test.

Similarly, creating a separate test folder segregating from the source folder ensured that the codes were more manageable and there was ease in distinction.

1. Relevancy and usefulness of the guidelines for good unit testing practice

Ans: The provided guidelines for good unit testing practice were relevant and exceptionally useful. The arrange, act, assert pattern enforces arranging objects , and setting them as per necessity, acting on the object and assert the expectation. One assert per test enforces clarity. The guideline for keeping it short and visible reasons very well as well. It also helps in keeping it maintainable and readable.

1. Level of easiness in achieving comprehensive test coverage. And, proportion of programming effort allocation for testing.

Ans: It is moderately challenging to achieve comprehensive test coverage. The issue being determining whether challenging or simplistic features are covered or is remaining in the coverage. Simply allotting the area or percentage of the test coverage rather than the difficulty level can be a daunting task.

In regards to programming effort for testing, 1:1 ratio between development and tests should be allotted to testing, thus about 45% to 50% of the effort should be allocated for testing. However, it also depends upon the familiarity with the code, experience and complexity of the task.

1. Effectiveness of dynamic testing compared to static code review in terms of discovering defects.

Ans: Dynamic testing is important for discovery of the defects. Dynamic testing is effective as it assess the dynamic behaviour of the code and it fulfils the validation process of testing. It highly focuses on identifying bugs unlike static testing that focuses on preventing defects. The intent of its use is to examine the test item so it is easier to accurately correlate errors, isolate them and fix the underlying problem. Thus, dynamic testing is more effective compared to static code review in respect to discovering defects as static testing aligns with preventing defects rather than finding and fixing the defects. The features of dynamic testing is that helps in determining bottlenecks in the program by identifying bugs.