Unit Testing Lab Report - CompSci280

UPI: mlew835 — ID: 589870827

August 2019

1 How did you go about completing each task and how does each solution work?

All my tests aim to follow the Arrange, Act, Assert pattern. In the maths module I have split my tests into different types. There are tests for integers, rational numbers, binary numbers, and hexadecimal numbers. This is to provide a range of test cases. I call the add function on these inputs and then use the AssertEqual function to see if the add method is providing the correct output. In regards to modifying the add function, I have written it so it checks to see if the base is 10. If it is 10 it returns the result as is, otherwise it calls on the convert base function to convert the result. After the conversion the result is returned.

For the Fib function, I adjusted it so that the function stopped its recursion once the count was at 1. If it was to go to 0 I noticed it was actually iterating through a sequence of 6 numbers and returning a value ahead.

For the conversion function, I choose two different bases and used the exact same process as the add tests to check the conversion was correct.

For the testing of the logger class I call on the logger methods to pass a message to my target class which it stores in a instance variable. I then checked that the value stored in the instance variable was the string I was expecting. As I discuss in the next question, I didn't fully understand the code for this question.

2 What difficulties did you face while attempting these tasks?

One of the main difficulties I encountered was Task Three. Unlike the other questions I wasn't sure what this question was asking me to do. I didn't understand what a target was, and how to test the logger class or how it really

works. After reading Piazza posts I think my understanding is somewhat better. The target acts as a stub (replacing what the logger might normally write to) and we use the target to test if the logger is passing its messages correctly. However, I still don't think I have a good understanding of how the code works for this question. I followed very closely Behzad's code on Piazza without fully understanding what was going on. I feel its very important to understand what you're doing so I didn't fully enjoy this process.

3 What did you learn from the tasks?

One of the things I learnt was that it is important to ensure your unit tests are actually correct before running them. I found that the values I set for comparison in the unit tests were actually wrong! During the unit tests I shouldn't be fixing the tests themselves!

I also got more famillar and confident with using the unittest framework as well as using the Arrange, Act and Assert structure in tests. I became more familar with the Eclispe Debugger and saw this as a good tool to replace running through the code by hand. I also think I leared the importance of starting projects earlier. If I had begun earlier I could have clarified my misunderstanding on Q3 by speaking to the lecturers and tutors.

4 How do you think this lab will help you in understanding software development?

Firstly, I am glad to have spent time working with the unittest module in Python. I had not known about this prior to the course and I hope to include it in my future projects. It is straightforward to use and I feel there is still a lot of functionality it provides I am yet to understand/explore.

It also has shown to me the importance of testing small units of code before combining other units together. Even within the small bit of code we dealt with in this assignment, there were major bugs which if they were integrated with other units of code could have lead to program faults. Software Development is much like a Jinga tower. The foundations have to be strong if the whole tower is stay standing.
.