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

For Task 1 and Task 2, completion of these tasks was pretty straight forward, really was just a matter of following the video on the Lab info page on Piazza. Following the Arrange, Act, Assert layout.

Task 1:

- Just had us writing simple test cases for adding two numbers together (using the add function), which wasn't really a problem.
- Then it had us debug some code (for the Fibonacci module), as it wasn't giving us the correct output, and make changes accordingly.

  Which after inspection was as simple as changing the return value; from "Third" to "Second" as returning the "Third" value originally was giving us the next number in the sequence.

  i.e. the 6th number in the Fibonacci sequence as when we are wanting the 5th – We would pass 'n' as we want the 'nth' number in the sequence but at first, we were getting the 'nth + 1' term.

  After these small changes our unit testing worked fine

Task 2

- Was writing some more simple test cases
- Making a small change to the convert_base function – '/' to '//'
- Editing the add function, so that it would take an extra optional parameter that would change the added number into another base (say from base 10 to base 16) by calling the convert_base function
- Then testing these changes again

Task 3

- More unit tests – this time testing a class as oppose to testing separate functions like task 1 and 2
- By far the most confusing and time-consuming task for me
- Found it hard to set the target
- Kept getting None

What difficulties did you face while attempting these tasks?

- Task 3 by was the hardest I found. Not that writing the tests was hard, just understanding how to pass the test a 'Target' is what I couldn't figure out.
- I kept getting an output of 'None' or just a class object at first

What did you learn from the tasks?

- How to write unit tests using the unit test library built into python.
- More emphasis on the importance
- How different assert methods work, and how for these cases one of the best asserts is the 'assertEqual' as it checks out input to see whether it matches our predefined expected output.

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

- This lab really emphasised the importance of testing and logging of changes made not necessarily only to do with software development, but in practice testing of any changes is always good
- It in general is very important to actively test your programs
- Every slight change should be tested, little by little
- You don't want to make too many changes without testing such that at first is working, but then after all the changes it suddenly breaks – it's very hard to go backwards
- Thus, logging is also very important when it comes to testing – as it helps us to keep track of all the changes we've made, making it easier for us to go backwards
- Another important idea – not really covered in this lab is the importance of making copies/versions