Reflective Summary

The peer-testing activity had various different outcomes, it was interesting and thought provoking to see the other ways students had implemented their code and the various solutions they had to problems I found in the course work task.

There are a few different changes I would make to my implementation of HashWordMap and HashWordMapTests after participating in the peer-testing task. There are areas I could of improved upon in my tests, rather than testing the whole method itself with a JUnit test, I could of tested individual methods and functions, this is something I done using print statements and live testing but made no record of. Using unit testing is much more efficient in hindsight, it would of allowed me to implement other parts of the code and simply run the JUnit tests I already had to check if my new implementation effected the other methods in any negative way, rather than going back and writing more print statements. I also believe this would of helped me visualise the different tasks each method was responsible for in respect to the larger program.

There were a few issues I encountered while testing my peers methods, firstly there were a few members who did not participate in the peer testing, this stopped me from running any tests on their code, this also reduced the sample size that I was able to learn from. A number of my peers did not implement the required methods, this gave me errors and null values when I ran my tests, another peers code had errors in some of their methods, in particular there was an arithmetic error meaning it returned 0 instead of the hash code, this gave me an assertion error as it returned against the value my JUnit test had calculated it should.

When running tests for the hash code, one of my peers implementation was incorrect meaning there were returning the ascii value using the Horner method of each character and not the hash code, for example, the ascii code of ‘a’ is 97 and the hash code is 97%13 = 6. This meant that while testing I got assertion errors as the JUnit was expecting the value after it had been hashed and not the ascii code. Finally, one peer had a math error with the Horner method, they were doing the calculation the wrong way around, resulting in the wrong characters being multiplied in a string of multiple characters.

Unfortunately I didn’t receive any helpful feedback on my code from the other peers in my group, as there were still some errors in my code I only got feedback in general telling me that they received errors when testing, rather than testing on the methods that were completed without any errors.

In conclusion the peer testing was some what helpful in developing my programming skills, it would have been more effective had all the peers participated and given constructive feedback, but it did allow me to see other more creative ways to solve certain problems that I wouldn’t of thought of had I not participated in the peer-testing task.