Reflective Summary

The peer-testing activity had various outcomes, it was interesting and thought provoking to see the other ways students had implemented their code and the different solutions they had to problems I found in the course work task. I also noticed some tips or smart tricks that I will now use in the future to save myself time and code duplication.

There are a few different changes I would make to my implementation of the Journey class, after participating in the peer-testing I noticed one of my peers had used lots of helper methods rather than having large methods that do multiple things like my implementation, this is something I would change, a method should do one thing and do it well, rather than lots of smaller task which can become confusing, for example with the airtime problem where you had to adding padding of 0’s if a flight flew through the night (e.g., Arrival: 05:00 – Departure: 13:00 = -8000 + 2400 = 16 hour flight). Having a separate helper method to the do padding for me then simply calling this helper method would be much more readable and also allow me to use this method if for example the code was going to be expanded in the future to incorporate train journeys or shipping traffic. Another addition I would make to my code and will now use more often in the future is the use of ‘for each’ loops rather than the traditional for index loops, these are much simpler when dealing with hash sets.

There were a few issues I encountered while testing my peer’s programs, firstly there were a couple of members who did not participate in the peer testing or uploaded their code with all the methods unimplemented. This meant I was unable to run any meaningful tests on their code, I did however comment on their implementation of part A. Another peer was trying to convert a String to type Airport, and I was given an error, as well as override errors as their FlyingPlanner was not abstract and could not override the abstract method leastTimeMeetUp in their IFlyingPlannerPartC interface. One peer had implemented the methods but had not returned the correct information, so the implementation was correct, but the test failed as the method did not return what it should have, this was seen in their leastCost method in their FlyingPlanner Class.

One of my peers had a very long and complicated populate method in FlyingPlanner but they had not included any comments so it was difficult to understand and follow their thought process line by line, I commented on this explaining that they could include line by line commenting as well as java doc comments to allow a user to see the methods parameters, any exceptions it throws and what it returns, as a glance.

Unfortunately, I have not received any feedback on my code from the peers in my group, I believe this may be due to exams.

In conclusion the peer testing was somewhat helpful in developing my programming skills, I have learned small tips and tricks to use in the future, but it would have been more effective had all of my peers participated and given feedback on my solutions.