Portfolio 2 Brief Questions

Q1. What would be the advantages and disadvantages if FlightQueue were to implement the

Queue interface?

*Then the queue would be sorted by an attribute (priority) and it would not require as complex methods due to the inbuilt methods.*

Q2. Which method(s) should be abstract in AbstractFlightQueue and which should remain

concrete? Explain your choice.

*The only value that should be abstract is the whole class as all the methods should be abstract to make the other classes (PriorityFlightQueue1, PriorityFlightQueue2, NormalFlightQueue) a lot simpler for when manipulating the instances of the classes in FlightTest002.*

Q3. Explain the algorithm that you implemented for PriorityFlightQueue2, in particular evaluate

the performance and the order of the algorithm using Big O notation

*For PriorityFlight2 I used a while loop. This is so I can find the exact spot as to where the newly added Flight instance can be placed. There is a Boolean statement within the while loop using the Boolean method &&(AND). This makes sure the index initialized at 0 remains below the total size of the flights list. Then there is a mathematical operator of <=(Less than or equal to), this makes sure the attribute priority is either below or equal to the current priority of the index of the list. When this loop breaks the index variable will be at the right value at which the new Flight belongs. We can then add the Flight to the list at that point which will then increment all the indexes of the Flights after the current index by 1.*

Q4. Does the flight queue appear sorted when you print it? If not why not, see the JavaDocs for

a hint? Is it evident that queue was sorted when the flights are landed?

*The flight queue does not look sorted initially however after landing a Flight from the queue and redisplaying the full queue it appears sorted.*