**REFLECTIVE REPORT – Pair Project**

The individual project tasked me to write a program that, given a start, city outputs a series of flights that take a passenger from the start city to the destination city. The main challenge with this project was the syntax due to the recent exposure to the c++ programming language. The project started with studying the CSV files that contained the database. With the aid of flowcharts, the different components of the program were visualized, and other needed parts were added as building the code proceeded. I then followed up by creating a project directory and a package and figured out the organization of the various files I needed. The concepts of defining classes, fields, constructors, methods, and error handling to create blueprints of the objects were applied. Prior to this, we made use of the ifstream() to open and read from the CSV files; to help temporarily hold and manipulate the available data, vectors and maps were used. This aspect of the project became challenging as to identifying the best algorithm to perform the task and how to integrate it with the data structure I had created for the program to use. I decided to use the breadth-first search algorithm for the search aspect of the task to find all possible routes (flights) from a source airport to a destination airport. The last task performed was to make the program take an input file and generate the result in an output file using the ofstream(). For this project, I sought help from faculty interns to have an idea of the best data structures to use for storing and manipulating the data. I also used knowledge from another class, Introduction to Artificial Intelligence, in the algorithm implementation, which turned out to be the most challenging part of the project because I found it difficult to connect it to the data structure used. In all, I think this project made me go out of my comfort zone in terms of the c++ syntax and also in terms of being able to read from and write to a file and store its data in a data structure with optimal time and space complexities.