REFLECTIVE REPORT ON FINAL PROJECT

INTERMEDIATE COMPUTER PROGRAMMING – C++

Student: Halidu Mubasir

ID: 51332024

Instructor: Dr. Robert Sowah

Faculty Intern: Owusu-Banahene Osie

Date: November 29, 2022

For this project, I worked in a team with Abubakari Sadik Osman. Our task was translating the code we had previously written in java to solve the problem into C++. Even though there were a few differences in our individual implementations of the java code, the overall approach was generally similar. We started by creating C++ classes for all three CSV files. i.e., we created a class for each of airport, airline, and route. We also had classes to store the collection of each of the airline, airport, and route objects. We used appropriate data structures in the project, such as unordered\_map, vectors, deque, and sets to store the objects. We created methods that read the CSV files and obtained the fields/attributes of each of the classes. We then used the attributes to create objects of the classes.

To represent the problem, which is to find a route or a series of routes between two places, we created a class called FindRoute. In this class, we had methods such as sourceAiport which gets the initial airport based on the initial city and country specified. We also had getSuccessors method, which generates successive airports based on the initial airport. It generates all possible airports a flight can go to based on the routes. We also had the isDestination method which takes in airport object as an argument and checks to see if that airport is the destination one wants to go. In the same class, we implemented a search method which uses the breadth-first search algorithm to find and return a solution path. To return the solution path, we had a Node class, where we kept track of airport and parent airports, i.e., airports that led to successive airports.

We faced a lot of challenges during the process of translating the code. First, we had difficulties with debugging because we could not understand some of the error messages. Then, the issues with pointers. With pointers, we did not have better understanding with how they work. During the implementation, we used pointers and anytime we had an error message we attach an asterisk sign to see if the error message goes. Through that process, we used the pointers till we began to fully understand how they work. By the end of the project, we could use pointers correctly without getting errors. I learnt a lot of C++ programming through the project especially the use of pointers. Because of the number of classes and methods we had in the code, object oriented programming has become part of my programming skills.