

REFLECTIVE PAPER

The scope of this project was to write a programmer which, given a start city (e.g., Accra, Ghana) and a destination city (e.g., Winnipeg, Canada), outputs a series of flights that take a passenger from the start city to the destination city. The basic requirement of this program is to determine a valid route between the two cities and go further to implement an optimal route. Identifying the classes, methods and data structures to be used were not much of a hurdle due to the work that had already gone into developing the Java header files and frameworks for the code to run. However, working with C++ introduced many opportunities as well as a few complexities. The major challenge we faced was with debugging the various modules of our program to appropriately import every file explicitly where it was required in other sections of the code.

To satisfy all requirements of the project we utilized the following C++ libraries including the appropriate header files (Airport.h, Route.h etc):

#include <iostream> : for providing basic input and output services for the program in C++

#include <vector> : For storing elements of similar data types in a memory space that can grow dynamically

#include <fstream> : ofstream and ifstream as an ostream for reading and writing to and from a file.

#include <sstream> : stringstream Class for performing insertions and extraction by parsing inputs and converting strings to numbers and numbers to strings.

#include <string> :To store a stream of user inputted characters in the object memory

using namespace std : for identifier scoping and declaration to prevent name conflicts in the project.

To begin we first identified the major functional units of code that were needed to build the successful program. With a detailed pseudocode / commenting approach we listed out the preliminary sections in order of priority – starting by declaring all primary variables. Using String Array data structure we grouped the Data on the Source city, source country , destination city , destination country. For the Route determination aspect we also declared variables for source – destination airports, also stored in a String Array.

Moving on, we explored the list of airports in the provided database that had been traversed, keeping track of the airports that have been added to Route. Then instantiating objects for the

source and destination airport “SourceAirport” and “Destination Airport” respectively as well as the variable for storing the String name of the airport that is closest to the destination airport.

Once we had all these primary variables declared our next step was to initialize them, and traverse the database to extract relevant data and increment variable points, j to find the route first as shown below:

```
//Finding the route

# While min is not null:

# Increment j by 1

# if min is not the destination airport:

    #Create a list of all the destination Airports from min using the findAllAirports() method

    #Iterate through the list of destination airports

    #Reassign min to the destination airport which is the closest to the destination Airport

    # Add min to route at index j

#else if min is the destination airport:

    # add min to route at index j

    # Print found solution

    # iterate and print route

    # return
```

From this project, one major lesson I learnt was to always define the scope of the project and the requirements of the expected solution in order to have a high-level overview of the code. This way it becomes less cumbersome working with c++ libraries and classes. Finally, I also learnt that I must have a good reason for the kind of programming language I choose to implement a project, as per the needs of the program in terms of performance, speed, efficiency and library support.