Assignment 2 777488

Name: Hussein Madan Migadde

Student Number: 27403645

I, the undersigned, hereby declare that this is my own and personal work, except where the work(s) or publications of others have been acknowledged by means of reference techniques. I have read and understood Tutorial Matter- Study Guide, Lessons 1 – 29 regarding technical and presentation requirements, referencing techniques and plagiarism.

DATE: 29th March 2025

A program that simulates a flight booking system.

INPUT

#include <iostream>

#include <string>

#include <vector>

#include <iomanip>

#include <cctype>

using namespace std;

// Constants

const int NUM\_SEATS = 50;

const int FIRST\_CLASS\_ROWS = 4;

const double ECONOMY\_COST = 1600.00;

const double FIRST\_CLASS\_MULTIPLIER = 1.20;

// Flight structure to hold flight details

struct Flight {

string departureTime;

vector<bool> seats;

int bookings;

Flight(string time) : departureTime(time), seats(NUM\_SEATS, false), bookings(0) {}

};

// Function prototypes

void displayMenu(const vector<Flight>& flights);

void displaySeating(const Flight& flight);

int getSeatNumber(const Flight& flight);

void bookSeat(Flight& flight, int seatNumber);

void displayBookingTicket(const string& fullName, const Flight& flight, int seatNumber);

void displayBookingSummary(const vector<Flight>& flights);

int main() {

// Initialize flight times

vector<Flight> flights;

flights.push\_back(Flight("7:00"));

flights.push\_back(Flight("9:00"));

flights.push\_back(Flight("11:00"));

flights.push\_back(Flight("13:00"));

flights.push\_back(Flight("15:00"));

string fullName;

char continueBooking;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\* \*" << endl;

cout << "\* Welcome to COS1511 Flight Booking System \*" << endl;

cout << "\* \*" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "Enter full name\n";

getline(cin, fullName);

do {

displayMenu(flights);

int choice;

cout << "Choose the time by entering the option number from the displayed" << "list: (1-5): ";

cin >> choice;

// Validate input range 1-5

while (choice < 1 || choice > 5) {

cout << "Incorrect option! Please choose from 1 - 5. ";

cin >> choice;

}

Flight& selectedFlight = flights[choice - 1];

displaySeating(selectedFlight);

int seatNumber = getSeatNumber(selectedFlight);

bookSeat(selectedFlight, seatNumber);

displayBookingTicket(fullName, selectedFlight, seatNumber);

selectedFlight.bookings++;

cout << "Do you want to make another booking? (Y/N): ";

cin >> continueBooking;

cin.ignore();

} while (toupper(continueBooking) == 'Y'); // Continue booking if the user // enters 'Y' or 'y'

displayBookingSummary(flights); // Display the total number of bookings for // each flight time

cout << "\nThank you for using the COS1511 Flight Booking System!\n";

return 0;

}

// Function to display the flight menu

void displayMenu(const vector<Flight>& flights) {

cout << "\nThe available travel times for flights are:\n";

cout << setw(5) << "Option" << setw(10) << "Depart" << setw(10) << "Arrive\n";

cout << setw(25) << setfill('-') << "" << endl;

cout << setfill(' ');

for (size\_t i = 0; i < flights.size(); ++i) {

int departureHour = stoi(flights[i].departureTime.substr(0, 2));

int arrivalHour = (departureHour + 2) % 24;

string arrivalTime = (arrivalHour < 10 ? "0" : "") + to\_string(arrivalHour) + ":30";

cout << setw(5) << to\_string(i + 1) + "." << setw(10) << flights[i].departureTime << setw(10) << arrivalTime << endl;

}

}

// Function to display the seating arrangement

void displaySeating(const Flight& flight) {

cout << "\nThe available seats are as follows:\n";

cout << " First class (R1920.00)\n";

char rowLabel = 'A';

for (int i = 0; i < NUM\_SEATS; ++i) {

int seatNum = (i % 6) + 1;

string seatLabel = string(1, rowLabel) + to\_string(seatNum);

cout << " | " << (flight.seats[i] ? "\*\*" : seatLabel) << " ";

if (seatNum == 3) {

cout << " | ------ ";

}

if (seatNum == 6) {

cout << " |\n";

rowLabel++;

if (rowLabel == 'E') {

cout << " | Economy class (R1600.00)\n";

}

}

}

}

// Function to get the seat number from the user

int getSeatNumber(const Flight& flight) {

string seatInput;

bool validSeat = false;

int seatNumber = 0;

while (!validSeat) {

cout << "\nSeats that are already taken are indicated with an"

<< " asterisk";

cout << "\nPlease key in a seat number to choose a seat (e.g., A1): ";

cin >> seatInput;

cin.ignore();

// Validate input format (e.g., A1, B2, etc.)

if (seatInput.length() != 2) {

cout << "Invalid format! Please enter a seat number like A1, B2, " << " etc.\n";

continue;

}

char row = toupper(seatInput[0]);

int column = seatInput[1] - '0';

// Validate row (A-I) and column (1-6)

if (row < 'A' || row > 'I') {

cout << "Invalid row! Please choose from A to I.\n";

continue;

}

if (column < 1 || column > 6) {

cout << "Invalid column! Please choose from 1 to 6.\n";

continue;

}

// Calculate the seat index in the vector

int rowIndex = row - 'A';

seatNumber = rowIndex \* 6 + (column - 1);

// Check if seat number exceeds total seats (0-49)

if (seatNumber >= NUM\_SEATS) {

cout << "Invalid seat! This seat does not exist.\n";

continue;

}

// Check if the seat is already booked

if (flight.seats[seatNumber]) {

cout << "That seat is already booked. Please choose another.\n";

} else {

validSeat = true;

}

}

return seatNumber + 1;

}

// Function to book the seat

void bookSeat(Flight& flight, int seatNumber) {

flight.seats[seatNumber - 1] = true;

}

// Function to display the booking ticket

void displayBookingTicket(const string& fullName, const Flight& flight, int seatNumber) {

double ticketPrice;

string seatClass;

// Convert seat number to letter + number format

char row = 'A' + (seatNumber - 1) / 6;

int column = ((seatNumber - 1) % 6) + 1;

string seatLabel = string(1, row) + to\_string(column);

if (seatNumber <= FIRST\_CLASS\_ROWS \* 6) {

ticketPrice = ECONOMY\_COST \* FIRST\_CLASS\_MULTIPLIER;

seatClass = "First class";

} else {

ticketPrice = ECONOMY\_COST;

seatClass = "Economy";

}

int departureHour = stoi(flight.departureTime.substr(0, 2));

int arrivalHour = (departureHour + 2) % 24;

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Travel ticket for FLIGHT";

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << left << setw(20) << "Name:" << setw(30) << fullName

<< "Travel Ticket class\t: " << seatClass << endl;

cout << left << setw(20) << "Departure:" << setw(30) << "Johannesburg"

<< "Seat Number\t\t: " << seatLabel << endl;

cout << left << setw(20) << "Destination:" << setw(30) << "Cape Town"

<< "Departure Time\t: " << flight.departureTime << endl;

cout << left << setw(20) << "" << setw(30) << ""

<< "Arrival Time\t\t: " << setw(2) << setfill('0') << arrivalHour << ":30" << setfill(' ') << endl;

cout.setf(ios::fixed);

cout.precision(2);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "Amount: R" << ticketPrice << " Thank you for booking with"

<< " COS1511."

<< " Your travel agent for this query is Hussein Madan"

<< endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

// Function to display the booking summary

void displayBookingSummary(const vector<Flight>& flights) {

cout << "\nBooking Summary:\n";

cout << "------------------------------------" << endl;

for (size\_t i = 0; i < flights.size(); ++i) {

cout << "Number of bookings made for " << flights[i].departureTime << ": " << flights[i].bookings << " \n";

}

}

**OUTPUT**



















