**Object Oriented Programming**

**Assignment 03**

Name: Muhammad Shahbaz Hassan

Roll Number: 24K-0884

Class: BCS-2C

**---- Code ----**

**#include<iostream>**

**#include<vector>**

**#include<fstream>**

**#include<string>**

**using namespace std;**

**class Person;**

**class Driver ;**

**class Student;**

**class Faculty;**

**class Transporter;**

**class Vehicle;**

**class Route;**

**class Seat;**

**class Booking;**

**class Route{**

**int id;**

**string start;**

**string end;**

**float distance;**

**public:**

**Route(){}**

**Route(int id,string a, string b, int c):id(id),start(a),end(b),distance(c){}**

**friend ostream& operator<<(ostream& out,Route& r){**

**out<<"\nRoute ID: "<<r.id<<endl;**

**out<<"Start Destination: "<<r.start<<endl;**

**out<<"End Destination: "<<r.end<<endl;**

**out<<"Distance: "<<r.distance<<endl;**

**return out;**

**}**

**void display\_route(){**

**cout<<start<<" -----> "<<end<<endl;**

**}**

**int get\_id(){return id;}**

**};**

**class Person{**

**int id;**

**string name;**

**int age;**

**string type;**

**bool paid=false;**

**public:**

**Person(){}**

**Person(int id,string a, int b,string d):id(id),name(a),age(b),type(d){}**

**virtual void print(ostream& out){**

**out<<"\nID: "<<id<<endl;**

**out<<"Name: "<<name<<endl;**

**out<<"Age: "<<age<<endl;**

**out<<"Type: "<<type<<endl;**

**if(type!="Driver"){**

**if(paid==true) out<<"Payment: Paid"<<endl;**

**else out<<"Payment: Not Paid"<<endl;**

**}**

**}**

**friend ostream& operator<<(ostream& out,Person& object){**

**object.print(out);**

**return out;**

**}**

**void set\_name(string a){name=a;}**

**void set\_age(int a){age=a;}**

**void set\_paid(bool a){paid=a;}**

**int get\_id(){return id;}**

**bool get\_pay\_status(){return paid;}**

**string get\_type(){return type;}**

**string get\_name(){return name;}**

**virtual int get\_fare(){return -1;};**

**virtual bool get\_has\_vehicle(){return false;};**

**virtual void add\_vehicle(Vehicle\* v){}**

**virtual void add\_seat(Seat\* s){}**

**virtual bool get\_is\_bus\_ac(){return false;}**

**virtual void set\_is\_bus\_ac(bool a){}**

**};**

**class Seat{**

**int number;**

**bool available=true;**

**Person\* person;**

**string type;**

**public:**

**Seat(){}**

**Seat(int a, string c):number(a),type(c),person(nullptr){}**

**void set\_available(bool a){available=a;}**

**bool get\_available(){return available;}**

**string get\_type(){return type;}**

**void add\_person(Person\* p){person=p;}**

**int get\_number(){return number;}**

**friend ostream& operator<<(ostream& out, Seat& s){**

**out<<"\nSeat Number: "<<s.number<<endl;**

**if(s.available==true){**

**out<<"Available= Yes"<<endl;**

**out<<"No Person has booked it yet";**

**}**

**else{**

**out<<"Available: No"<<endl;**

**out<<"Person Info: ";**

**s.person->print(out);**

**}**

**return out;**

**}**

**};**

**class Vehicle{**

**int id;**

**string type;**

**string number;**

**bool is\_AC;**

**int student\_seats;**

**int faculty\_seats;**

**int seats;**

**vector<Seat\*> seating;**

**Route\* route;**

**Person\* driver;**

**public:**

**Vehicle(Person\* drive):driver(drive){}**

**Vehicle(int id,string type,string a, bool b, int s, int ss, int fs,Person\* drive):id(id),type(type),number(a),is\_AC(b),seats(s),student\_seats(ss),faculty\_seats(fs),route(nullptr),driver(drive){**

**for(int i=0;i<faculty\_seats;i++){**

**seating.push\_back(new Seat(i+1,"Faculty"));**

**}**

**for(int i=0;i<student\_seats;i++){**

**seating.push\_back(new Seat(i+1,"Student"));**

**}**

**}**

**bool get\_AC(){return is\_AC;}**

**string get\_type(){return type;}**

**int get\_id(){return id;}**

**int get\_total\_seats(){return seats;}**

**int get\_student\_seats(){return student\_seats;}**

**int get\_faculty\_seats(){return faculty\_seats;}**

**void add\_route(Route\* r){route=r;}**

**Route\* get\_route(){return route;}**

**vector<Seat\*> get\_seat(){return seating;}**

**void print(ostream& out){**

**out<<"\nID: "<<id<<endl;**

**out<<"Type: "<<type<<endl;**

**if(is\_AC==true) out<<"AC: Yes"<<endl;**

**else cout<<"AC: No"<<endl;**

**out<<"Total Seats: "<<seats<<endl;**

**out<<"Student Seats: "<<student\_seats<<endl;**

**out<<"Faculty Seats: "<<faculty\_seats<<endl;**

**out<<"----- Driver Info: ";**

**driver->print(out);**

**cout<<"----- Route Info: ";**

**if(route!=nullptr) out<<\*route;**

**else out<<"No Route Added"<<endl;**

**}**

**friend ostream& operator<<(ostream& out,Vehicle& object){**

**object.print(out);**

**return out;**

**}**

**int count\_remain\_student\_seats(){**

**int count=0;**

**for(int i=0;i<seating.size();i++){**

**if(seating[i]->get\_type()=="Student"&&seating[i]->get\_available()==true){**

**count++;**

**}**

**}**

**return count;**

**}**

**int count\_remain\_faculty\_seats(){**

**int count=0;**

**for(int i=0;i<seating.size();i++){**

**if(seating[i]->get\_type()=="Faculty"&&seating[i]->get\_available()==true){**

**count++;**

**}**

**}**

**return count;**

**}**

**};**

**class Bus:public Vehicle{**

**public:**

**Bus(Person\* drive):Vehicle(drive){}**

**Bus(int id, string a, bool b, int ss,Person\* drive):Vehicle(id,"Bus",a,b,52,ss,52-ss,drive){}**

**};**

**class Coaster:public Vehicle{**

**public:**

**Coaster(Person\* drive):Vehicle(drive){}**

**Coaster(int id, string a, bool b, int ss,Person\* drive):Vehicle(id,"Coaster",a,b,32,ss,32-ss,drive){}**

**};**

**class Driver:public Person{**

**int License;**

**Vehicle\* vehicle;**

**bool has\_vehicle;**

**public:**

**Driver(){}**

**Driver(int id,string a, int b, int c):Person(id,a,b,"Driver"),License(c),vehicle(nullptr),has\_vehicle(false){}**

**void print(ostream& out) override{**

**Person::print(out);**

**cout<<"License: "<<License<<endl;**

**if(has\_vehicle==false) cout<<"Vehicle: Not Assigned"<<endl;**

**else cout<<"Vehicle: Assigned"<<endl;**

**}**

**bool get\_has\_vehicle() override {return has\_vehicle;}**

**void add\_vehicle(Vehicle\* v){**

**vehicle=v;**

**has\_vehicle=true;**

**}**

**};**

**class Student:public Person{**

**Vehicle\* vehicle;**

**Seat\* seat;**

**bool is\_bus\_ac;**

**public:**

**static int fare;**

**Student (){}**

**Student (int id,string a, int b):Person(id,a,b,"Student"),vehicle(nullptr),is\_bus\_ac(false){}**

**void print(ostream& out) override{**

**Person::print(out);**

**if(vehicle!=nullptr){**

**out<<"Vehicle Id: "<<vehicle->get\_id()<<endl;**

**if(seat!=nullptr){**

**out<<"Seat Number Booked: "<<seat->get\_number()<<endl;**

**}**

**}**

**}**

**int get\_fare() override {**

**return fare;**

**}**

**void add\_vehicle(Vehicle\* v)override {vehicle=v;}**

**void add\_seat(Seat\* s)override {seat=s;}**

**bool get\_is\_bus\_ac()override {return is\_bus\_ac;}**

**void set\_is\_bus\_ac(bool a)override {is\_bus\_ac=a;}**

**};**

**class Faculty:public Person{**

**Vehicle\* vehicle;**

**Seat\* seat;**

**bool is\_bus\_ac;**

**public:**

**static int fare;**

**Faculty (){}**

**Faculty (int id,string a, int b):Person(id,a,b,"Faculty"),vehicle(nullptr),is\_bus\_ac(false){}**

**void print(ostream& out) override{**

**Person::print(out);**

**if(vehicle!=nullptr) {**

**out<<"Vehicle Id: "<<vehicle->get\_id()<<endl;**

**if(seat!=nullptr){**

**out<<"Seat Number Booked: "<<seat->get\_number()<<endl;**

**}**

**}**

**}**

**int get\_fare() override {**

**return fare;**

**}**

**void add\_vehicle(Vehicle\* v)override {vehicle=v;}**

**void add\_seat(Seat\* s)override {seat=s;}**

**bool get\_is\_bus\_ac()override {return is\_bus\_ac;}**

**void set\_is\_bus\_ac(bool a)override {is\_bus\_ac=a;}**

**};**

**int Student::fare=1500;**

**int Faculty::fare=1000;**

**class Booking{**

**int number;**

**Seat\* seat;**

**Person\* person;**

**Vehicle\* vehicle;**

**Route\* route;**

**public:**

**Booking():seat(nullptr),person(nullptr),vehicle(nullptr),route(nullptr){}**

**Booking(int i,Seat\* s,Person\* p, Vehicle\* v):number(i),seat(s),person(p),vehicle(v){**

**route=vehicle->get\_route();**

**}**

**friend ostream& operator<<(ostream& out, Booking& b){**

**out<<"\nBooking Number: "<<b.number<<endl;**

**out<<"Person ID: "<<b.person->get\_id()<<endl;**

**out<<"Person Name: "<<b.person->get\_name()<<endl;**

**out<<"Seat Number: "<<b.seat->get\_number()<<endl;**

**out<<"Vehicle ID: "<<b.vehicle->get\_id()<<endl;**

**out<<"Route ID: "<<b.route->get\_id()<<endl;**

**b.route->display\_route();**

**return out;**

**}**

**};**

**class Transporter{**

**string name;**

**vector<Route\*> routes;**

**vector<Person\*> drivers;**

**vector<Vehicle\*> vehicles;**

**vector<Booking\*>booking;**

**public:**

**Transporter(){}**

**Transporter(string a):name(a){}**

**string get\_name(){return name;}**

**vector<Person\*> get\_drivers(){return drivers;}**

**vector<Vehicle\*> get\_vehicle(){return vehicles;}**

**vector<Route\*> get\_route(){return routes;}**

**vector<Booking\*> get\_booking(){return booking;}**

**void add\_driver(string name, int age, int license){**

**drivers.push\_back(new Driver(drivers.size()+10,name,age,license));**

**}**

**void add\_vehicles(string number,bool ac ,int ss, int d,Person\* driver){**

**if(d==1){**

**vehicles.push\_back(new Bus(vehicles.size()+20,number,ac,ss,driver));**

**}**

**else{**

**vehicles.push\_back(new Coaster(vehicles.size()+20,number,ac,ss,driver));**

**}**

**}**

**void add\_route(string a, string b,float c){**

**routes.push\_back(new Route(routes.size()+30,a,b,c));**

**}**

**void add\_booking(Seat\* s,Person\* p, Vehicle\* v){**

**booking.push\_back(new Booking(booking.size()+1,s,p,v));**

**}**

**};**

**class File\_Handler{**

**fstream file;**

**public:**

**File\_Handler(){**

**fstream test("User.txt",ios::in);**

**if(!test.is\_open()){**

**file.open("User.txt",ios::out);**

**file.close();**

**}**

**test.close();**

**}**

**void add\_person(vector<Person\*> person){**

**remove("User.txt");**

**file.open("User.txt",ios::out);**

**for(int i=0;i<person.size();i++){**

**file<<\*person[i];**

**}**

**file.close();**

**}**

**void give\_data(vector<Person\*>\* person){**

**file.open("User.txt");**

**string str;**

**while(getline(file,str)){**

**if(str.size()==0){**

**string name,type;**

**int age,id;**

**getline(file,str);**

**if(str.find("ID: ")!=-1){**

**int n=str.find(":");**

**str=str.substr(n+2,str.size());**

**id=stoi(str);**

**}**

**getline(file,str);**

**if(str.find("Name: ")!=-1){**

**int n=str.find(":");**

**str=str.substr(n+2,str.size());**

**name=str;**

**}**

**getline(file,str);**

**if(str.find("Age: ")!=-1){**

**int n=str.find(":");**

**str=str.substr(n+2,str.size());**

**age=stoi(str);**

**}**

**getline(file,str);**

**if(str.find("Type: ")!=-1){**

**int n=str.find(":");**

**str=str.substr(n+2,str.size());**

**type=str;**

**}**

**string extra;**

**getline(file,extra);**

**if(extra.size()!=0){**

**int n=extra.find(":");**

**extra=extra.substr(n+2,extra.size());**

**}**

**if(type=="Student"){**

**person->push\_back(new Student(id,name,age));**

**}**

**else if(type=="Faculty"){**

**person->push\_back(new Faculty(id,name,age));**

**}**

**if(extra=="Paid"){**

**(\*person)[person->size()-1]->set\_paid(true);**

**}**

**}**

**}**

**file.close();**

**}**

**};**

**int main(){**

***//declarations***

**vector<Person\*> person;**

**vector<Transporter\*> transporter;**

**File\_Handler file;**

**int a;**

**transporter.push\_back(new Transporter("Nadeem"));**

**transporter.push\_back(new Transporter("Zulfiqar"));**

**file.give\_data(&person);**

***// main menu loop***

**while(1){**

**cout<<"\n----- MAIN MENU -----\n1. Users(Student or Faculty)\n2. Transporter\n3. Exit\nEnter the Corresponding number for the option: ";**

**cin>>a;**

***//User Menu***

**if(a==1){**

**int b;**

**while(1){**

**cout<<"\n----- MAIN MENU > USER -----\n1. Add a User\n2. Display All Users\n3. Update a User Info\n4. Pay a fee\n5. Book a Seat\n9. Exit\nEnter the Corresponding number for the option: ";**

**cin>>b;**

***//Adding a user***

**if(b==1){**

**string name,type;**

**int age,c;**

**cout<<"\nYou want to Register:\n1. Student ?\n2. Faculty ?\nEnter the Corresponding number to the option: ";**

**cin>>c;**

**if(c==1||c==2){**

**cout<<"Enter the Name of the User: ";**

**cin>>name;**

**cout<<"Enter the Age of the User: ";**

**cin>>age;**

**if(c==1){**

**person.push\_back(new Student(person.size()+1000,name,age));**

**cout<<"New Student Added Successfully"<<endl;**

**file.add\_person(person);**

**}**

**else{**

**person.push\_back(new Faculty(person.size()+1000,name,age));**

**cout<<"New Faculty Added Successfully"<<endl;**

**file.add\_person(person);**

**}**

**}**

**else{**

**cout<<"Invalid Option | Selection terminated"<<endl;**

**}**

**}**

***//Display all users***

**else if(b==2){**

**if(person.size()!=0){**

**for(int i=0;i<person.size();i++){**

**cout<<\*person[i];**

**cout<<endl;**

**}**

**}**

**else{**

**cout<<"No User Added Yet"<<endl;**

**}**

**}**

***//Update a user info***

**else if(b==3){**

**int i,c;**

**bool check=false;**

**cout<<"Enter the id of the user whose info to be updated: ";**

**cin>>c;**

**for(i=0;i<person.size();i++){**

**if(person[i]->get\_id()==c){**

**check=true;**

**break;**

**}**

**}**

**if(check==true){**

**cout<<"Update:\n1. Name ?\n2. Age ?\nEnter the Corresponding number for the option: ";**

**cin>>c;**

**if(c==1){**

**string name;**

**cout<<"Enter the new name: ";**

**cin>>name;**

**person[i]->set\_name(name);**

**cout<<"Updated Info: "<<endl;**

**cout<<\*person[i];**

**}**

**else if(c==2){**

**int age;**

**cout<<"Enter the new Age: ";**

**cin>>age;**

**person[i]->set\_age(age);**

**cout<<"Updated Info: "<<endl;**

**cout<<\*person[i];**

**}**

**else{**

**cout<<"Wrong Option | Updation Terminated"<<endl;**

**}**

**}**

**else{**

**cout<<"No User of this Id Found"<<endl;**

**}**

**}**

***// pay the fee***

**else if(b==4){**

**int id,i,ac,fare;**

**bool check=false;**

**cout<<"Enter the ID of the Person whose seat is to be registered: ";**

**cin>>id;**

**for(i=0;i<person.size();i++){**

**if(person[i]->get\_id()==id){**

**check=true;**

**break;**

**}**

**}**

**if(check==true){**

**cout<<"You want AC bus seat? (0 for No and 1 for Yes): ";**

**cin>>ac;**

**if(ac==0){**

**cout<<"You need to Pay the Fee of "<<person[i]->get\_fare()<<endl;**

**cout<<"Press 1 to pay the fee: ";**

**cin>>fare;**

**if(fare==1){**

**person[i]->set\_paid(true);**

**person[i]->set\_is\_bus\_ac(false);**

**cout<<"Fee Paid"<<endl;**

**file.add\_person(person);**

**}**

**else cout<<"Invald Output"<<endl;**

**}**

**else if(ac==1){**

**cout<<"You need to Pay the Fee of "<<person[i]->get\_fare()+2000<<endl;**

**cout<<"Press 1 to pay the fee: ";**

**cin>>fare;**

**if(fare==1){**

**person[i]->set\_paid(true);**

**person[i]->set\_is\_bus\_ac(true);**

**cout<<"Fee Paid"<<endl;**

**file.add\_person(person);**

**}**

**else cout<<"Invald Output"<<endl;**

**}**

**else cout<<"Invalid Output"<<endl;**

**}**

**else cout<<"No Person of this ID found"<<endl;**

**}**

***//register for a seat***

**else if(b==5){**

**int id,i,j,transport;**

**bool check=false,check2=false;**

**bool  zul=false, nad=false;**

**cout<<"Enter the ID of the Person whose seat is to be registered: ";**

**cin>>id;**

**for(i=0;i<person.size();i++){**

**if(person[i]->get\_id()==id){**

**check=true;**

**break;**

**}**

**}**

**if(check==true){**

**try{**

**if(person[i]->get\_pay\_status()==false){**

**throw runtime\_error("Booking Failed | No Payment ");**

**}**

**if(transporter[0]->get\_vehicle().size()!=0||transporter[1]->get\_vehicle().size()!=0){**

**if(transporter[0]->get\_route().size()!=0||transporter[1]->get\_route().size()!=0){**

**cout<<"The list of Transporters with Active Vehicles: ";**

***// nadeem printing***

**for(int j=0;j<transporter[0]->get\_vehicle().size();j++){**

**if(transporter[0]->get\_vehicle()[j]->get\_route()!=nullptr){**

**nad=true;**

**break;**

**}**

**}**

**if(transporter[0]->get\_route().size()!=0&&nad==true){**

**cout<<"NADEEM TRANSPORTER: "<<endl;**

**for(int j=0;j<transporter[0]->get\_vehicle().size();j++){**

**if(person[i]->get\_is\_bus\_ac()==transporter[0]->get\_vehicle()[j]->get\_AC()){**

**cout<<\*transporter[0]->get\_vehicle()[j]<<endl;**

**}**

**}**

**}**

**else if(transporter[0]->get\_route().size()!=0&&nad==false) cout<<" Nadeem Transporter have No Vehicles with defined Routes | Give Routes to the Vehicles"<<endl;**

**else cout<<"Nadeem Transporter have No Routes at all"<<endl;**

***// zulfiqar printing***

**for(int j=0;j<transporter[1]->get\_vehicle().size();j++){**

**if(transporter[1]->get\_vehicle()[j]->get\_route()!=nullptr){**

**zul=true;**

**break;**

**}**

**}**

**if(transporter[1]->get\_route().size()!=0&&zul==true){**

**cout<<"ZULFIQAR TRANSPORTER: "<<endl;**

**for(int j=0;j<transporter[1]->get\_vehicle().size();j++){**

**if(person[i]->get\_is\_bus\_ac()==transporter[1]->get\_vehicle()[j]->get\_AC()){**

**cout<<\*transporter[0]->get\_vehicle()[j]<<endl;**

**}**

**}**

**}**

**else if(transporter[1]->get\_route().size()!=0&&zul==false) cout<<" Zulfiqar Transporter have No Vehicles with defined Routes | Give Routes to the Vehicles"<<endl;**

**else cout<<"Zulfiqar Transporter have No Routes at all"<<endl;**

***// seat booking***

**if(zul==true||nad==true){**

**vector<int> arr;**

**if(nad==true){**

**arr.push\_back(0);**

**}**

**if(zul==true){**

**arr.push\_back(1);**

**}**

**if(arr.size()==1){**

**if(arr[0]==0){**

**transport=0;**

**}**

**if(arr[0]==1) {**

**transport=1;**

**}**

**}**

**else if(arr.size()>0){**

**cout<<"Choose one of the Transporters whom to register to\n1. Nadeem\n2. Zulfiqar\nEnter: ";**

**cin>>transport;**

**if(transport==0||transport==1){}**

**else cout<<"Invalid Option"<<endl;**

**}**

**if(transport==0||transport==1){**

**int j;**

**check2=false;**

**cout<<"Enter the Id of the Vehicle you want to register to: ";**

**cin>>id;**

**for(j=0;j<transporter[transport]->get\_vehicle().size();j++){**

**if(transporter[transport]->get\_vehicle()[j]->get\_id()==id){**

**check2=true;**

**break;**

**}**

**}**

**if(check2==true){**

**cout<<"Everything good"<<endl;**

**check=false;**

**if(person[i]->get\_type()=="Student"){**

**if(transporter[transport]->get\_vehicle()[j]->get\_student\_seats()>0){**

**for(int k=0;k<transporter[transport]->get\_vehicle()[j]->get\_seat().size();k++){**

**if(transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->get\_available()==true&&transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->get\_type()==person[i]->get\_type()){**

**person[i]->add\_seat(transporter[transport]->get\_vehicle()[j]->get\_seat()[k]);**

**transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->add\_person(person[i]);**

**person[i]->add\_vehicle(transporter[transport]->get\_vehicle()[j]);**

**transporter[transport]->add\_booking(transporter[transport]->get\_vehicle()[j]->get\_seat()[k],person[i],transporter[transport]->get\_vehicle()[j]);**

**transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->set\_available(false);**

**cout<<"Seat Booked Successfully"<<endl;**

**cout<<\*transporter[transport]->get\_booking()[transporter[transport]->get\_booking().size()-1];**

**check=true;**

**break;**

**}**

**}**

**if(check==false) cout<<"No Seat Available"<<endl;**

**}**

**}**

**else if(person[i]->get\_type()=="Faculty"){**

**if(transporter[transport]->get\_vehicle()[j]->get\_faculty\_seats()>0){**

**for(int k=0;k<transporter[transport]->get\_vehicle()[j]->get\_seat().size();k++){**

**if(transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->get\_available()==true&&transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->get\_type()==person[i]->get\_type()){**

**person[i]->add\_seat(transporter[transport]->get\_vehicle()[j]->get\_seat()[k]);**

**transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->add\_person(person[i]);**

**person[i]->add\_vehicle(transporter[transport]->get\_vehicle()[j]);**

**transporter[transport]->add\_booking(transporter[transport]->get\_vehicle()[j]->get\_seat()[k],person[i],transporter[transport]->get\_vehicle()[j]);**

**transporter[transport]->get\_vehicle()[j]->get\_seat()[k]->set\_available(false);**

**cout<<"Seat Booked Successfully"<<endl;**

**cout<<\*transporter[transport]->get\_booking()[transporter[transport]->get\_booking().size()-1];**

**check=true;**

**break;**

**}**

**}**

**if(check==false) cout<<"No Seat Available"<<endl;**

**}**

**}**

**else cout<<"No "<<person[i]->get\_type()<<" Seat available in this "<<transporter[transport]->get\_vehicle()[j]->get\_type() <<" now"<<endl;**

**}**

**else cout<<"No Vehicle of this ID found"<<endl;**

**}**

**}**

**}**

**else cout<<"Transporters have No Routes Assigned"<<endl;**

**}**

**else cout<<"Transporters have No Active Vehicles right now"<<endl;**

**}**

**catch(runtime\_error& e){**

**cout<<e.what()<<endl;**

**}**

**}**

**else cout<<"No Person of this ID found"<<endl;**

**}**

**else if(b==9) break;**

**else cout<<"Invalid Number"<<endl;**

**file.add\_person(person);**

**}**

**}**

***// Transporter Menu***

**else if(a==2){**

**int b;**

**int transport;**

**string transport\_name;**

**while(1){**

**cout<<"\n----- MAIN MENU > TRANSPORTER -----\n1. Nadeem\n2. Zulfiqar\n3. Exit\nEnter the Corresponding number for the option: ";**

**cin>>b;**

**if(b==1||b==2){**

**if(b==1) {**

**transport=0;**

**transport\_name="NADEEM";**

**}**

**else{**

**transport=1;**

**transport\_name="ZULFIQAR";**

**}**

**while(1){**

**int c;**

**cout<<"\n----- MAIN MENU > TRANSPORTER > "<<transport\_name<<" -----\n1. Add a Driver\n2. Display All Drivers\n3. Add a Vehicle\n4. Display All Vehicles\n5. Add a Route\n6. Display All Routes\n7. Added Route to a Vehicle\n8. Display all bookings\n9. Exit\nEnter the Corresponding number for the option: ";**

**cin>>c;**

***//Add a driver***

**if(c==1){**

**string name;**

**int age, license;**

**cout<<"Enter the Name of Driver: ";**

**cin>>name;**

**cout<<"Enter the Age of the Driver: ";**

**cin>>age;**

**cout<<"Enter the License number of the Driver: ";**

**cin>>license;**

**transporter[transport]->add\_driver(name,age,license);**

**cout<<"Driver Added Successfully"<<endl;**

**cout<<\*transporter[transport]->get\_drivers()[transporter[transport]->get\_drivers().size()-1];**

**}**

***// display all drivers***

**else if(c==2){**

**if(transporter[transport]->get\_drivers().size()!=0){**

**for(int i=0;i<transporter[transport]->get\_drivers().size();i++){**

**cout<<\*transporter[transport]->get\_drivers()[i]<<endl;**

**}**

**}**

**else cout<<"Drivers have not been added"<<endl;**

**}**

***// add vehicle***

**else if(c==3){**

**if(transporter[transport]->get\_drivers().size()!=0){**

**string number;**

**bool ac;**

**int d,ss,type,e,i;**

**cout<<"Which type of Vehicle you are adding:\n1. Bus\n2. Coaster\nEnter the Corresponding number for the option: ";**

**cin>>d;**

**if(d==1||d==2){**

**cout<<"Enter the Plate number of the Vehicle: ";**

**cin>>number;**

**if(d==1){**

**do{**

**cout<<"Enter the number of student seats out of 52(others are faculty ones): ";**

**cin>>ss;**

**if(52-ss<0) cout<<"Enter Again"<<endl;**

**}while(52-ss<0);**

**}**

**else{**

**do{**

**cout<<"Enter the number of student seats out of 32(others are faculty ones): ";**

**cin>>ss;**

**if(32-ss<0) cout<<"Enter Again"<<endl;**

**}while(32-ss<0);**

**}**

**do{**

**cout<<"Is the Bus AC ? (0 for No and 1 for Yes): ";**

**cin>>ac;**

**if(ac==0||ac==1) break;**

**else cout<<"Enter Again | Only 0 or 1 "<<endl;**

**}while(1);**

**cout<<"Now you need to choose the driver for the Vehicle\nHere is the list of these available: ";**

**for(i=0;i<transporter[transport]->get\_drivers().size();i++){**

**if(transporter[transport]->get\_drivers()[i]->get\_has\_vehicle()==false){**

**cout<<\*transporter[transport]->get\_drivers()[i];**

**}**

**}**

**cout<<"Enter the ID of the driver from the above list: ";**

**cin>>e;**

**bool check=false;**

**for(i=0;i<transporter[transport]->get\_drivers().size();i++){**

**if(transporter[transport]->get\_drivers()[i]->get\_has\_vehicle()==false&&transporter[transport]->get\_drivers()[i]->get\_id()==e){**

**check=true;**

**break;**

**}**

**}**

**if(check==true){**

**transporter[transport]->add\_vehicles(number,ac,ss,d,transporter[transport]->get\_drivers()[i]);**

**transporter[transport]->get\_drivers()[i]->add\_vehicle(transporter[transport]->get\_vehicle()[transporter[transport]->get\_vehicle().size()-1]);**

**cout<<"Vehicle added successfully"<<endl;**

**cout<<\*transporter[transport]->get\_vehicle()[transporter[transport]->get\_vehicle().size()-1];**

**}**

**else cout<<"No ID Found | Vehicle Adding Failed"<<endl;**

**}**

**else cout<<"Invalid Option"<<endl;**

**}**

**else cout<<"You do not have Drivers | Added them first"<<endl;**

**}**

***//display all vehicles***

**else if(c==4){**

**if(transporter[transport]->get\_vehicle().size()!=0){**

**for(int i=0;i<transporter[transport]->get\_vehicle().size();i++){**

**cout<<\*transporter[transport]->get\_vehicle()[i]<<endl;**

**}**

**}**

**else cout<<"Empty | Add Vehicles first"<<endl;**

**}**

***// Add routes***

**else if(c==5){**

**string start,end;**

**float distance;**

**cout<<"Enter the Start Location: ";**

**cin>>start;**

**cout<<"Enter the End Location: ";**

**cin>>end;**

**cout<<"Enter the Distance between them: ";**

**cin>>distance;**

**transporter[transport]->add\_route(start,end,distance);**

**cout<<"Route Added Successfully"<<endl;**

**cout<<\*transporter[transport]->get\_route()[transporter[transport]->get\_route().size()-1];**

**}**

***// display all routes***

**else if(c==6){**

**if(transporter[transport]->get\_route().size()!=0){**

**for(int i=0;i<transporter[transport]->get\_route().size();i++){**

**cout<<\*transporter[transport]->get\_route()[i]<<endl;**

**}**

**}**

**else cout<<"Empty | Add Routes first"<<endl;**

**}**

***// add route to a vehicle***

**else if(c==7){**

**if(transporter[transport]->get\_vehicle().size()!=0){**

**if(transporter[transport]->get\_route().size()!=0){**

**int id,id2,i,j;**

**bool check=false,check2=false;**

**cout<<"Enter the Vehicle to which the route is to be added: ";**

**cin>>id2;**

**for(j=0;j<transporter[transport]->get\_vehicle().size();j++){**

**if(transporter[transport]->get\_vehicle()[j]->get\_id()==id2){**

**check2=true;**

**break;**

**}**

**}**

**if(check2==true){**

**for(i=0;i<transporter[transport]->get\_route().size();i++){**

**cout<<\*transporter[transport]->get\_route()[i]<<endl;**

**}**

**cout<<"Choose from the above. Enter the ID of the route: ";**

**cin>>id;**

**for(i=0;i<transporter[transport]->get\_route().size();i++){**

**if(transporter[transport]->get\_route()[i]->get\_id()==id){**

**check=true;**

**break;**

**}**

**}**

**if(check==true){**

**transporter[transport]->get\_vehicle()[j]->add\_route(transporter[transport]->get\_route()[i]);**

**cout<<"Route Added Successfully";**

**}**

**else cout<<"No Route of this ID Found"<<endl;**

**}**

**else cout<<"No Vehicle Found"<<endl;**

**}**

**else cout<<"Add Routes First"<<endl;**

**}**

**else cout<<"Add Vehicles First"<<endl;**

**}**

***// display all booking***

**else if(c==8){**

**if(transporter[transport]->get\_booking().size()!=0){**

**for(int i=0;i<transporter[transport]->get\_booking().size();i++){**

**cout<<\*transporter[transport]->get\_booking()[i]<<endl;**

**}**

**}**

**else cout<<"Empty | Add booking first"<<endl;**

**}**

**else if(c==9) break;**

**else cout<<"Invalid Option"<<endl;**

**}**

**}**

**else if(b==3) break;**

**else cout<<"Invalid Option" <<endl;**

**}**

**}**

**else if(a==3) break;**

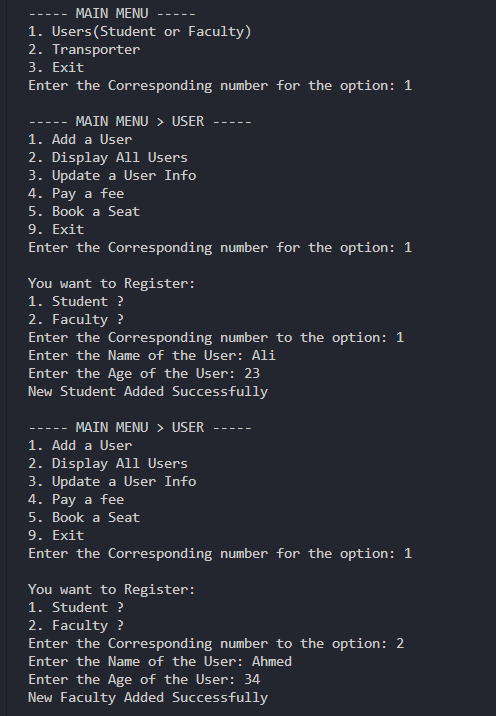
**else cout<<"Invaid Number"<<endl;**

**}**

**return 0;**

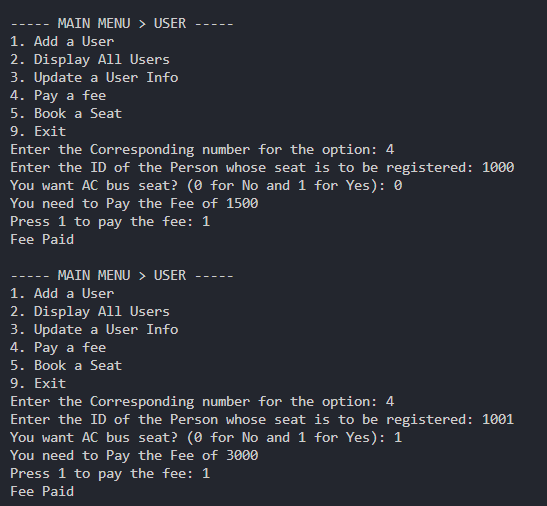
**}**

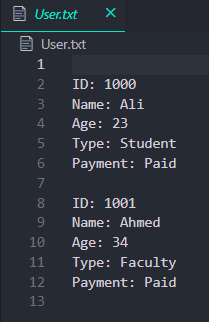
**---- Output Screenshots ----**

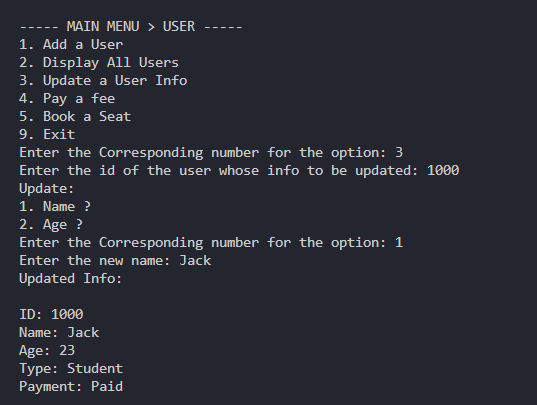
****

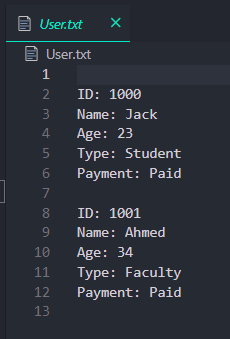
**A screenshot of a computer

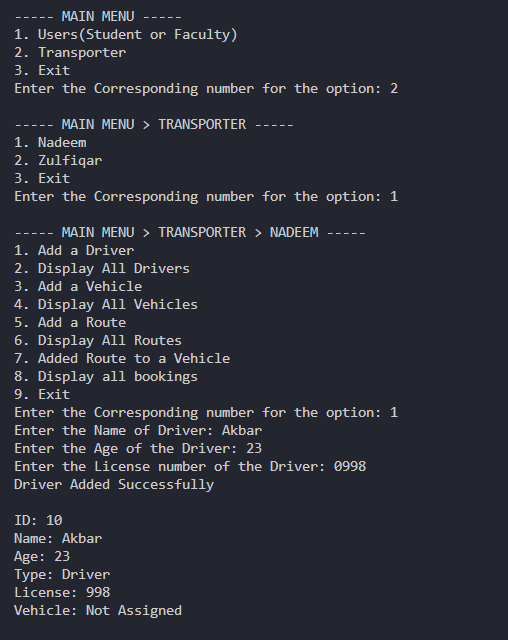
AI-generated content may be incorrect.**

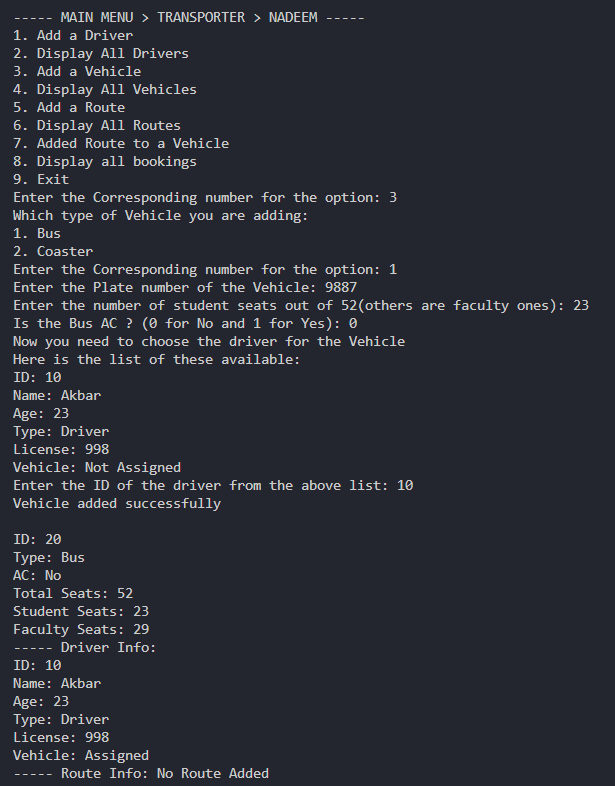
****

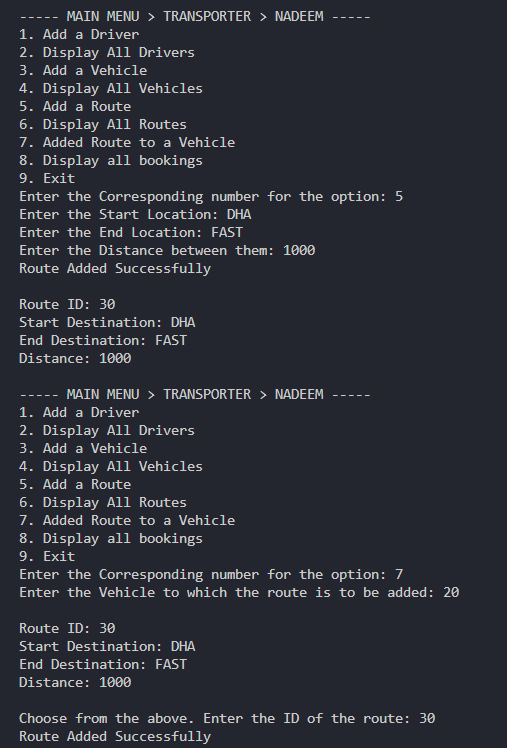
****

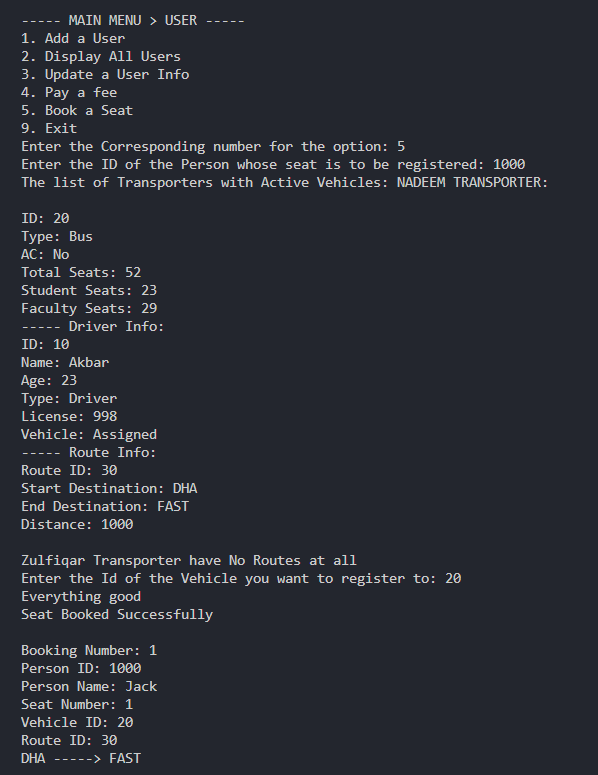
****

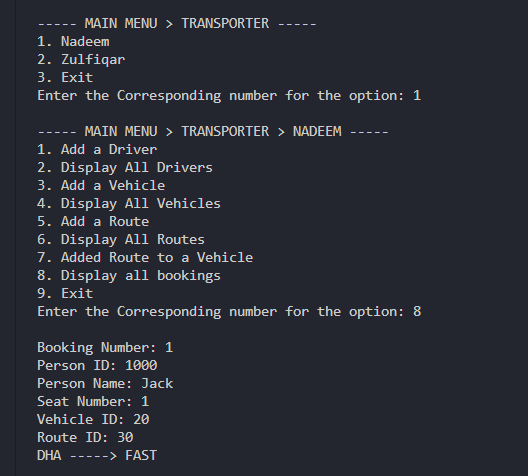
****

****

****

****

****

****