**BAZIL UDDIN KHAN**

**24K-0559**

**BSCS -2H**

**QUESTION 1).**

**CODE:**

**// 24K-0559 BAZIL-UDDIN-KHAN**

**#include <iostream>**

**using namespace std;**

**class Mentor;**

**class Student**

**{**

**private:**

**string StudentId;**

**string Name;**

**int Age;**

**string SportsInterests[100];**

**string MentorAssigned;**

**int i;**

**public:**

**Student()**

**{**

**StudentId = " ";**

**Name = " ";**

**Age =0;**

**for(int i =0; i < 100; i++)**

**{**

**SportsInterests[i] = " ";**

**}**

**MentorAssigned = " ";**

**i =0;**

**}**

**Student(string StudentId, string Name, int Age,string MentorAssigned)**

**{**

**this->StudentId = StudentId;**

**this->Name = Name;**

**this->Age = Age;**

**this->MentorAssigned = MentorAssigned;**

**}**

**void SetAttributes(string StudentId, string Name, int Age,string MentorAssigned)**

**{**

**this->StudentId = StudentId;**

**this->Name = Name;**

**this->Age = Age;**

**this->MentorAssigned = MentorAssigned;**

**}**

**void RegisterForMentorship(Mentor & m,string sportsinterest);**

**void UpdateSportsInterests()**

**{**

**int Priority;**

**cout << " Enter Your priority " << endl;**

**cin >> Priority;**

**string UpdateInterest;**

**cout << " Enter New Sports interest To be Updated " << endl;**

**cin >> UpdateInterest;**

**for(int i =0; i < 100; i++)**

**{**

**if(i+1 == Priority)**

**{**

**SportsInterests[i] = UpdateInterest;**

**}**

**}**

**}**

**void viewMentorDetails(Mentor & m);**

**string GetName()**

**{**

**return Name;**

**}**

**};**

**class Mentor**

**{**

**private:**

**string MentorId;**

**string Name;**

**string sportsExpertise[100];**

**int MaxLearners;**

**int CountLearners;**

**string \* AssignedLearners ;**

**public:**

**Mentor()**

**{**

**MentorId = " ";**

**Name = " ";**

**MaxLearners =0;**

**CountLearners =0;**

**AssignedLearners = nullptr;**

**}**

**Mentor(string MentorId,string Name,**

**int MaxLearners)**

**{**

**this->MentorId = MentorId;**

**this->Name = Name;**

**this->MaxLearners = MaxLearners;**

**CountLearners =0;**

**AssignedLearners = new string[MaxLearners];**

**}**

**void AssignLearner(Student & S);**

**void RemoveLearner();**

**void viewLearners()**

**{**

**cout << " Learners Are " << endl;**

**for(int i =0; i < CountLearners;i++)**

**{**

**cout << AssignedLearners[i] << " " << endl;**

**}**

**}**

**void ProvideGauidance()**

**{**

**string Message;**

**cout << " Enter A Question on which you want to fell You need Quaidance " << endl;**

**cin.ignore();**

**getline(cin,Message);**

**string Response;**

**cout << " Enter Response of Quaidance " << endl;**

**getline(cin,Response);**

**cout << " Reponse of Qs is " << Response << " Qs was " << Message << endl;**

**}**

**int GetCurrentCount()**

**{**

**return CountLearners;**

**}**

**int GetTotalcount()**

**{**

**return MaxLearners;**

**}**

**string GetName()**

**{**

**return Name;**

**}**

**void PrintInfo()**

**{**

**cout << " Mentor Name is " << Name << " " << " Mentor id is " << MentorId << " " << " Max quantity is " << MaxLearners << endl;**

**}**

**~Mentor()**

**{**

**delete[] AssignedLearners;**

**}**

**};**

**void Student :: viewMentorDetails(Mentor & m )**

**{**

**cout << " Mentor Details Are " << endl;**

**m.PrintInfo();**

**}**

**void Student :: RegisterForMentorship(Mentor & m,string sportsinterest)**

**{**

**if(m.GetCurrentCount() < m.GetTotalcount())**

**{**

**string studentid;**

**cout << " Enter Student Id like(21k-5678) " << endl;**

**cin >> studentid;**

**string name;**

**cout << " Enter Name " << endl;**

**cin >> name;**

**int age;**

**cout << " Enter Age " << endl;**

**cin >> age;**

**string mentorassigned;**

**cout << " Enter Your Mentor Assigned " << endl;**

**cin >> mentorassigned;**

**if(mentorassigned != m.GetName())**

**{**

**return;**

**}**

**SportsInterests[i++] = sportsinterest;**

**Student S(studentid,name,age,mentorassigned);**

**m.AssignLearner(S);**

**cout << " Student Succesfully Registered " << endl;**

**}**

**else**

**{**

**cout << " Sorry Its Full " << endl;**

**}**

**}**

**class Skill**

**{**

**private:**

**string SkillId;**

**string SkillName;**

**string Description;**

**public:**

**Skill()**

**{**

**SkillId = " ";**

**SkillName = " ";**

**Description = " ";**

**}**

**Skill(string SkillId,string SkillName**

**,string Description)**

**{**

**this->SkillId = SkillId;**

**this-> SkillName = SkillName;**

**this->Description = Description;**

**}**

**void InitalizeSkillDeatails(string SkillId,string SkillName,string Description)**

**{**

**this->SkillId = SkillId;**

**this->SkillName = SkillName;**

**this->Description = Description;**

**}**

**void ShowSkillDeatils()**

**{**

**cout << " Skill Id is " << SkillId << " " << " Skill Name is " << " " << SkillName << " " << " Description is " << Description << endl;**

**}**

**void UpdateSkillDescription(string NewDescription)**

**{**

**Description = NewDescription;**

**}**

**string Getskillname()**

**{**

**return SkillName;**

**}**

**};**

**void Mentor :: AssignLearner(Student & S)**

**{**

**if(CountLearners < MaxLearners)**

**{**

**AssignedLearners[CountLearners++] = S.GetName();**

**}**

**else**

**{**

**cout << " No space " << endl;**

**}**

**}**

**void Mentor :: RemoveLearner()**

**{**

**string RemovedLearner;**

**cout << " Enter Learner/Student to be removed " << endl;**

**cin >> RemovedLearner;**

**int Found = -1;**

**for(int i =0; i < CountLearners; i++)**

**{**

**if(AssignedLearners[i] == RemovedLearner)**

**{**

**Found =i;**

**break;**

**}**

**}**

**if(Found == -1)**

**{**

**cout << " Student Not Found " << endl;**

**}**

**else**

**{**

**for(int j =Found; j < CountLearners-1 ;j++ )**

**{**

**AssignedLearners[j] = AssignedLearners[j+1];**

**}**

**CountLearners--;**

**cout << " Succesfully Removed " << endl;**

**}**

**}**

**class Sport**

**{**

**private:**

**string SportId;**

**string Name;**

**string Description;**

**string \* RequiredSkills;**

**int Totalskills;**

**public:**

**Sport()**

**{**

**SportId = " ";**

**Name = " ";**

**Description = " ";**

**Totalskills =0;**

**RequiredSkills = new string[100];**

**}**

**Sport(string SportId,string Name,string Description)**

**{**

**this->SportId = SportId;**

**this->Name = Name;**

**this->Description = Description;**

**}**

**void Addskill(Skill s)**

**{**

**if(Totalskills < 100)**

**{**

**RequiredSkills[Totalskills] = s.Getskillname();**

**Totalskills++;**

**}**

**else**

**{**

**cout << " Skill are Reached " << endl;**

**}**

**}**

**void Removeskill(Skill s)**

**{**

**int Index = -1;**

**for(int i =0; i < Totalskills;i++)**

**{**

**if(RequiredSkills[i] != s.Getskillname())**

**{**

**Index =i;**

**break;**

**}**

**}**

**if(Index != -1)**

**{**

**for(int i =Index; i < Totalskills-1;i++)**

**{**

**RequiredSkills[i] = RequiredSkills[i+1];**

**}**

**Totalskills--;**

**}**

**else**

**{**

**cout << " Skill Not Found " << endl;**

**}**

**}**

**};**

**int main()**

**{**

**cout << " Welcome To Sports System "<< endl;**

**// Mentor Class**

**int Quantity;**

**string MentorId;**

**string MentorName;**

**cout << " Enter Mentor Id like (21k-4567) " << endl;**

**cin >> MentorId;**

**cout << " Enter Mentor Name " << endl;**

**cin >> MentorName;**

**cout << " Enter How Much Learners Can Mentor Teach " << endl;**

**cin >> Quantity;**

**Mentor mentor(MentorId,MentorName,Quantity);**

**// Skill Class**

**int TotalSkills;**

**cout << " Enter Total Skills of The Mentor " << endl;**

**cin >> TotalSkills;**

**Skill skill[TotalSkills];**

**for(int i =0; i < TotalSkills;i++)**

**{**

**string Skillid;**

**cout << " Enter Skill id " << endl;**

**cin >> Skillid;**

**string Skillname;**

**cout << " Enter Skill Name " << endl;**

**cin.ignore();**

**getline(cin,Skillname);**

**string Description;**

**cout << " Enter Skill Description " << endl;**

**getline(cin,Description);**

**skill[i].InitalizeSkillDeatails(Skillid,Skillname,Description);**

**}**

**for(int i =0; i < TotalSkills;i++)**

**{**

**skill[i].ShowSkillDeatils();**

**}**

**string Choice;**

**cout << " Do you want To update Your Skills. Enter in(Yes/No) " << endl;**

**cin >> Choice;**

**if(Choice == "Yes")**

**{**

**string Skillname;**

**cout << " Enter Skill Name To update " << endl;**

**cin >> Skillname;**

**for(int j =0; j < TotalSkills;j++)**

**{**

**if(skill[j].Getskillname() == Skillname)**

**{**

**string Newskill;**

**cout << " Enter Updated Skill " << endl;**

**cin >> Newskill;**

**skill[j].UpdateSkillDescription(Newskill);**

**}**

**}**

**skill->ShowSkillDeatils();**

**}**

**// Sport Class**

**int TotalSportsSkill;**

**cout << " Enter Total Sports Skills " << endl;**

**cin >> TotalSportsSkill;**

**Sport sport;**

**string Sportsinterest;**

**cout << " Enter Sports Interest " << endl;**

**cin >> Sportsinterest;**

**for(int i =0; i < TotalSportsSkill;i++)**

**{**

**string Choice;**

**cout << " Enter in Yes/No if you want to add skill " << endl;**

**cin >> Choice;**

**if(Choice == "Yes")**

**{**

**sport.Addskill(skill[i]);**

**}**

**string CHOICE;**

**cout << " Enter in Yes/No if you want to remove skill " << endl;**

**cin >> Choice;**

**if(CHOICE == "Yes")**

**{**

**sport.Removeskill(skill[i]);**

**}**

**string SportId;**

**cout << " Enter Sport Id " << endl;**

**cin >> SportId;**

**}**

**// Student Class**

**Student \* student = new Student[Quantity];**

**for(int i =0; i < Quantity; i++)**

**{**

**student[i].RegisterForMentorship(mentor,Sportsinterest);**

**}**

**student[0].viewMentorDetails(mentor);**

**mentor.viewLearners();**

**mentor.ProvideGauidance();**

**int Totalremove;**

**cout << " Enter Total Students To be Removed " << endl;**

**cin >> Totalremove;**

**if(Totalremove <= Quantity)**

**{**

**for(int i =0; i < Totalremove;i++)**

**{**

**mentor.RemoveLearner();**

**}**

**cout << " Updated Learners Now " << endl;**

**mentor.viewLearners();**

**}**

**else**

**{**

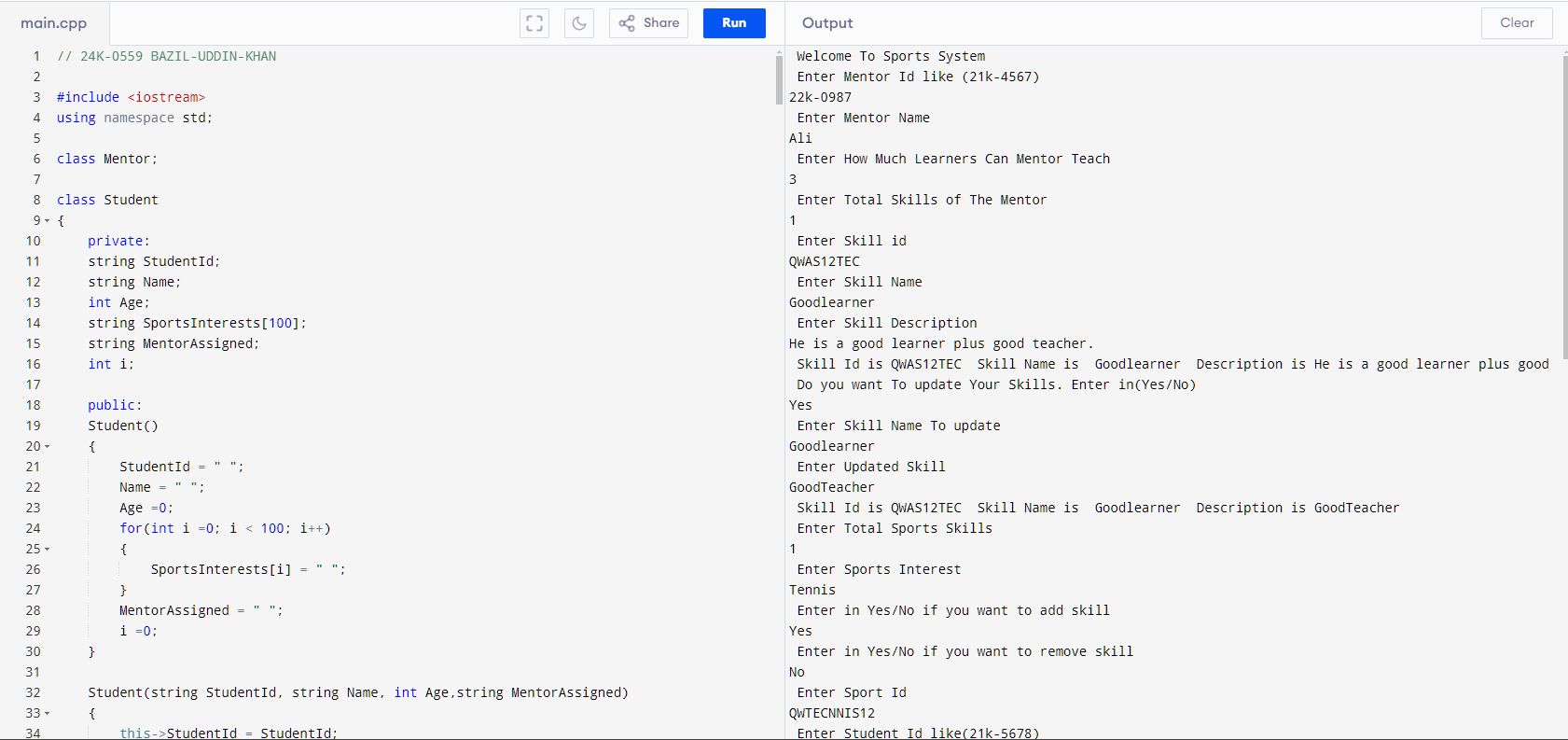
**cout << " Error You Are Trying To Remove More Students Than Are Learning " << endl;**

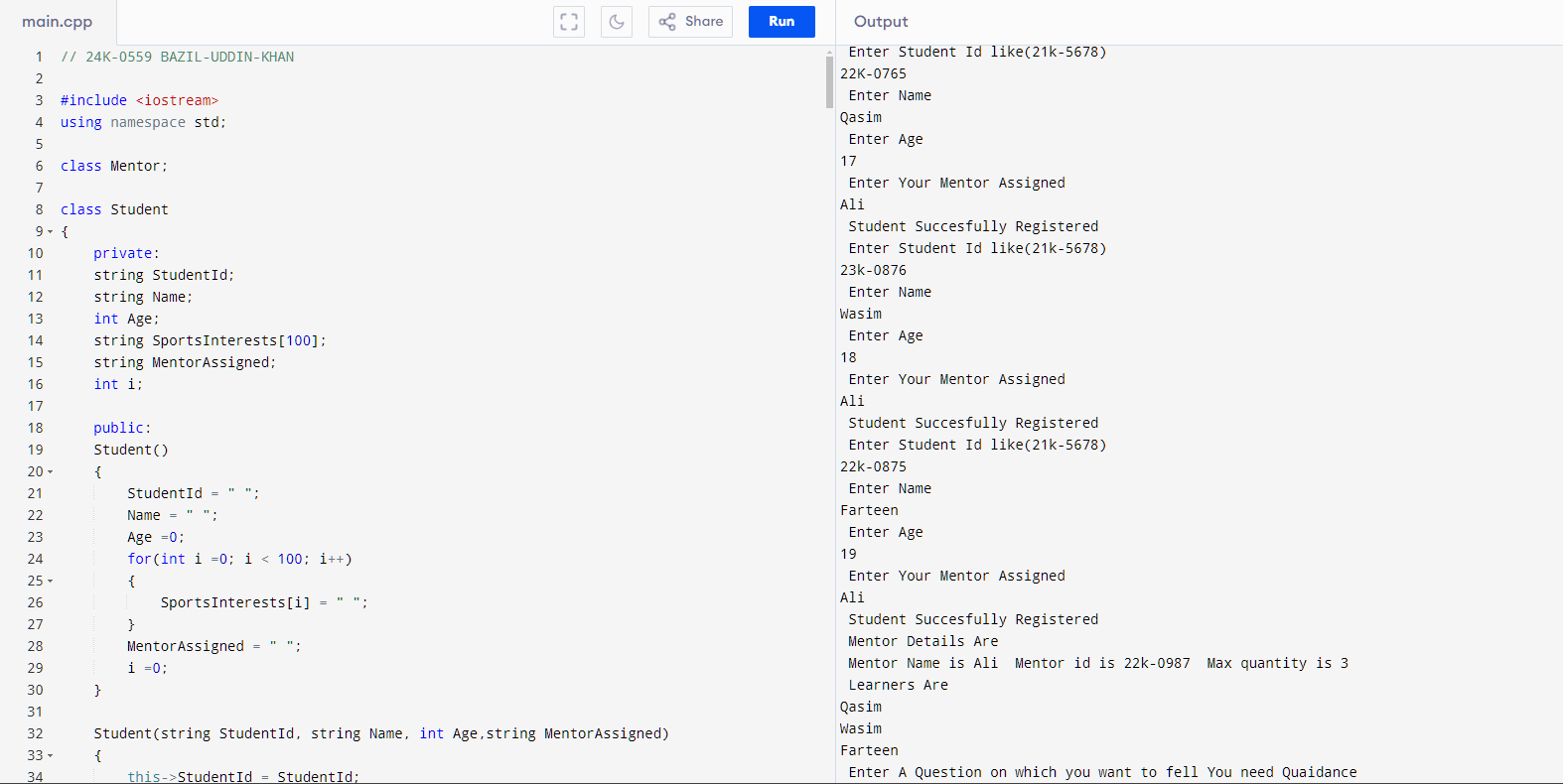
**}**

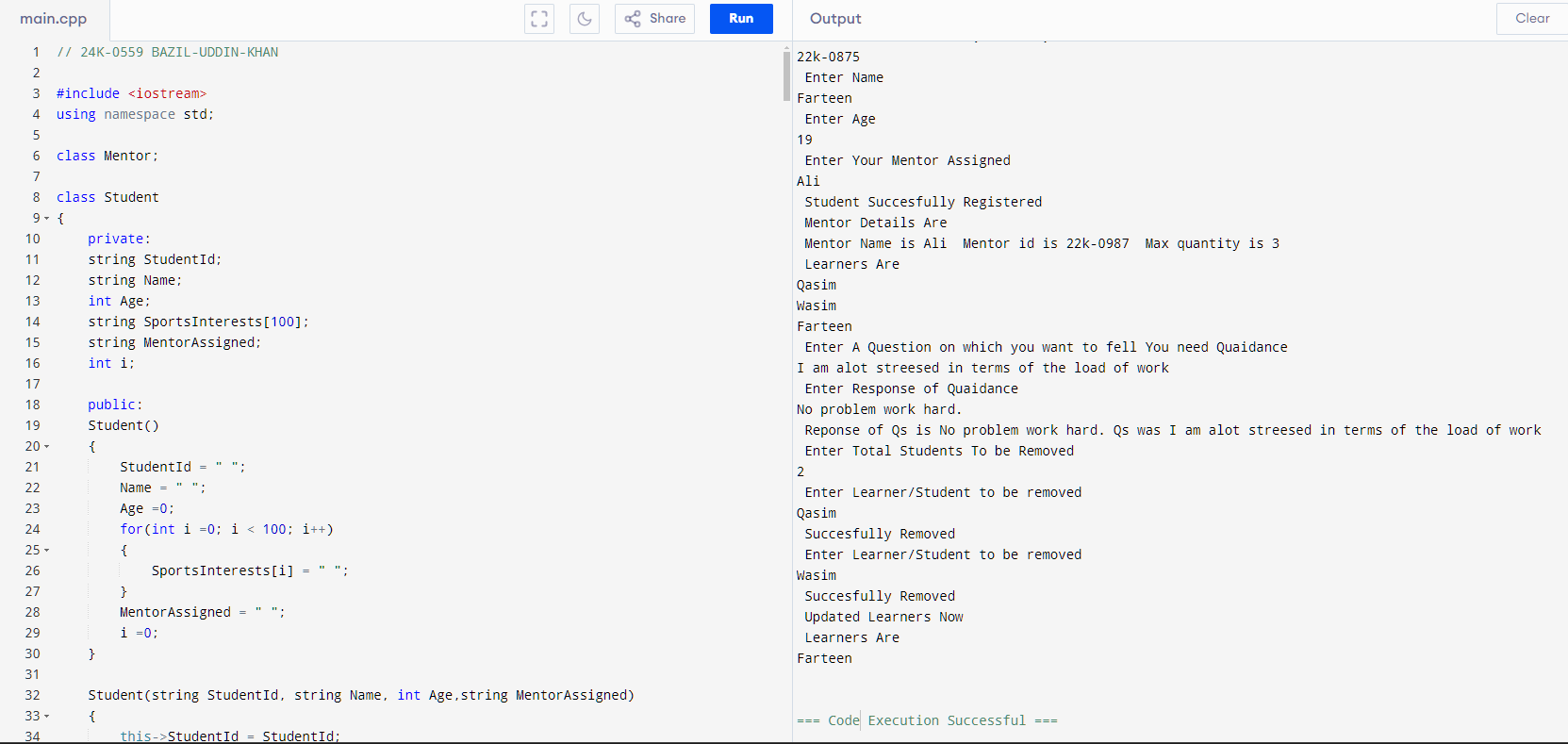
**return 0;**

**}**

**QUESTION 1 OUTPUT:**

****

****

****

**QUESTION2).**

**CODE:**

**// 24k-0559 BAZIL-UDDIN-KHAN**

**#include <iostream>**

**using namespace std;**

**class Ball**

**{**

**private:**

**int X;**

**int Y;**

**public:**

**Ball()**

**{**

**X =0;**

**Y =0;**

**}**

**void MethodMove(int Dx,int Dy)**

**{**

**this->X = Dx;**

**this->Y = Dy;**

**}**

**void SetCoordinates()**

**{**

**X = 0;**

**Y =0;**

**}**

**void GetPosition()**

**{**

**cout << " Current Position is (X,Y) " << X << " , " << Y << endl;**

**}**

**int GetX()**

**{**

**return X;**

**}**

**int GetY()**

**{**

**return Y;**

**}**

**};**

**class Robot**

**{**

**private:**

**string Name;**

**int TotalHits;**

**int RobotX;**

**int RobotY;**

**int Y ;**

**int X ;**

**Ball ball;**

**public:**

**Robot()**

**{**

**Name = " ";**

**TotalHits =0;**

**RobotX=0;**

**RobotY=0;**

**Y =0;**

**X =0;**

**}**

**int CallX()**

**{**

**return (ball.GetX());**

**}**

**int Cally()**

**{**

**return (ball.GetY());**

**}**

**string GetName()**

**{**

**return Name;**

**}**

**void SetCoordinates()**

**{**

**X = 0;**

**Y =0;**

**}**

**void SetName(string Name)**

**{**

**this->Name = Name;**

**}**

**int GetHits()**

**{**

**return TotalHits;**

**}**

**string UpDirection()**

**{**

**return "Up";**

**}**

**string DownDirection()**

**{**

**return "Down";**

**}**

**string LeftDirection()**

**{**

**return "Left";**

**}**

**string RightDirection()**

**{**

**return "Right";**

**}**

**void SetHits(int TotalHits)**

**{**

**this->TotalHits = TotalHits;**

**}**

**void HitBall(int &Ballx,int &Bally,const string &Direction,Ball &ball)**

**{**

**if(Direction == "Up")**

**{**

**this->Y = Y + Bally;**

**this->X = X + Ballx;**

**ball.MethodMove(X,Y);**

**this->RobotX = X;**

**this->RobotY = Y;**

**}**

**else if(Direction == "Down")**

**{**

**this->Y = Y - Bally;**

**this->X = X - Ballx;**

**ball.MethodMove(X,Y);**

**this->RobotX = X;**

**RobotY = Y;**

**}**

**else if(Direction == "Right")**

**{**

**this->X = X + Ballx;**

**ball.MethodMove(X,Y);**

**this->RobotX = X;**

**}**

**else if(Direction == "Left")**

**{**

**this->X = X - Ballx;**

**ball.MethodMove(X,Y);**

**this->RobotX = X;**

**}**

**else**

**{**

**cout << " Wrong Try " << endl;**

**}**

**TotalHits++;**

**}**

**};**

**class Goal**

**{**

**private:**

**int X;**

**int Y;**

**public:**

**Goal()**

**{**

**X =0;**

**Y =0;**

**}**

**Goal(int X, int Y)**

**{**

**this->X = X;**

**this->Y = Y;**

**}**

**void SetCoordinates()**

**{**

**this->X =0;**

**this->Y =0;**

**}**

**int GetTargetX()**

**{**

**return X;**

**}**

**int GetTargetY()**

**{**

**return Y;**

**}**

**int isGoalReached(int ballX, int ballY)**

**{**

**if(ballX == X && ballY == Y)**

**{**

**return 1;**

**}**

**else**

**{**

**cout << " Still Not Reached " << endl;**

**return 0;**

**}**

**}**

**};**

**class Team**

**{**

**private:**

**string Name;**

**Robot \* robot ;**

**int TeamNum;**

**int Teamhits;**

**public:**

**Team()**

**{**

**TeamNum =0;**

**robot = new Robot[2];**

**}**

**void SetTeam(string Name)**

**{**

**this->Name = Name;**

**}**

**void SetTeamHits(int Teamhits)**

**{**

**this->Teamhits= Teamhits;**

**}**

**void CallRobot(Team \* team)**

**{**

**Goal goal(3,3);**

**Ball ball;**

**for(int i =0; i < 2; i++)**

**{**

**string Name;**

**cout << " Enter Robot Name . (like this Qasim11 etc withno gaps) " << endl;**

**cin >> Name;**

**robot[i].SetName(Name);**

**int Teamhits =0;**

**while(1)**

**{**

**if(goal.isGoalReached(ball.GetX(),ball.GetY()) == 1)**

**{**

**cout << " Your Goal Has Reached " << endl;**

**ball.SetCoordinates();**

**cout << " Robot : " << robot[i].GetName() << " Reached Goal in " << robot[i].GetHits() << " Times " << endl;**

**robot[i].SetHits(Teamhits);**

**team[i].SetTeamHits(Teamhits);**

**TeamNum++;**

**break;**

**}**

**else**

**{**

**int EnterHitx;**

**cout << " Enter Hit x " << endl;**

**cin >> EnterHitx;**

**int EnterHity;**

**cout << " Enter Hit Y " << endl;**

**cin >> EnterHity;**

**string Direction;**

**cout << " Enter Direction(Up/Down/Left/Right) " << endl;**

**cin >> Direction;**

**robot[i].HitBall(EnterHitx,EnterHity,Direction,ball);**

**ball.GetPosition();**

**Teamhits++;**

**}**

**}**

**}**

**}**

**void SetTeam(int Teamhits)**

**{**

**this->Teamhits = Teamhits;**

**}**

**int GetTeam()**

**{**

**return Teamhits;**

**}**

**int TeamTotalHits(int i )**

**{**

**return (robot[i].GetHits());**

**}**

**~Team()**

**{**

**delete[] robot;**

**}**

**};**

**class Game**

**{**

**private:**

**Team Teamone;**

**Team Teamtwo;**

**Goal Goalobject;**

**Ball Ballobject;**

**public:**

**void startGame()**

**{**

**cout << " Start The Game " << endl;**

**}**

**void play(Team \*team)**

**{**

**team->CallRobot(team);**

**}**

**void DeclareWinner(Team \* team)**

**{**

**int TeamHitsone = team[0].GetTeam();**

**int TeamHitstwo = team[1].GetTeam();**

**Teamone.SetTeamHits(TeamHitsone);**

**Teamtwo.SetTeamHits(TeamHitstwo);**

**cout << " Teamone one hits " << TeamHitsone << endl;**

**cout << " Teamone Two hits " << TeamHitstwo << endl;**

**if(Teamone.GetTeam() < Teamtwo.GetTeam())**

**{**

**cout << " Congratulations Team 1 has Won with Total : " << Teamone.GetTeam() << " Hits " << endl;**

**}**

**else if(Teamtwo.GetTeam() < Teamone.GetTeam())**

**{**

**cout << " Congratulations Team 2 has Won with Total : " << Teamtwo.GetTeam() << " Hits " << endl;**

**}**

**else**

**{**

**cout << " Oh it's a tie " ;**

**}**

**}**

**};**

**int main()**

**{**

**Game GAME;**

**GAME.startGame();**

**Team \*team = new Team[2];**

**for(int i =0; i < 2; i++)**

**{**

**string Name;**

**cout << " Enter Team Name. (like this Qasim11 etc withno gaps)" << endl;**

**cin >> Name;**

**team[i].SetTeam(Name);**

**}**

**int Index =0;**

**for(int i =0; i < 1; i++)**

**{**

**GAME.play(team);**

**}**

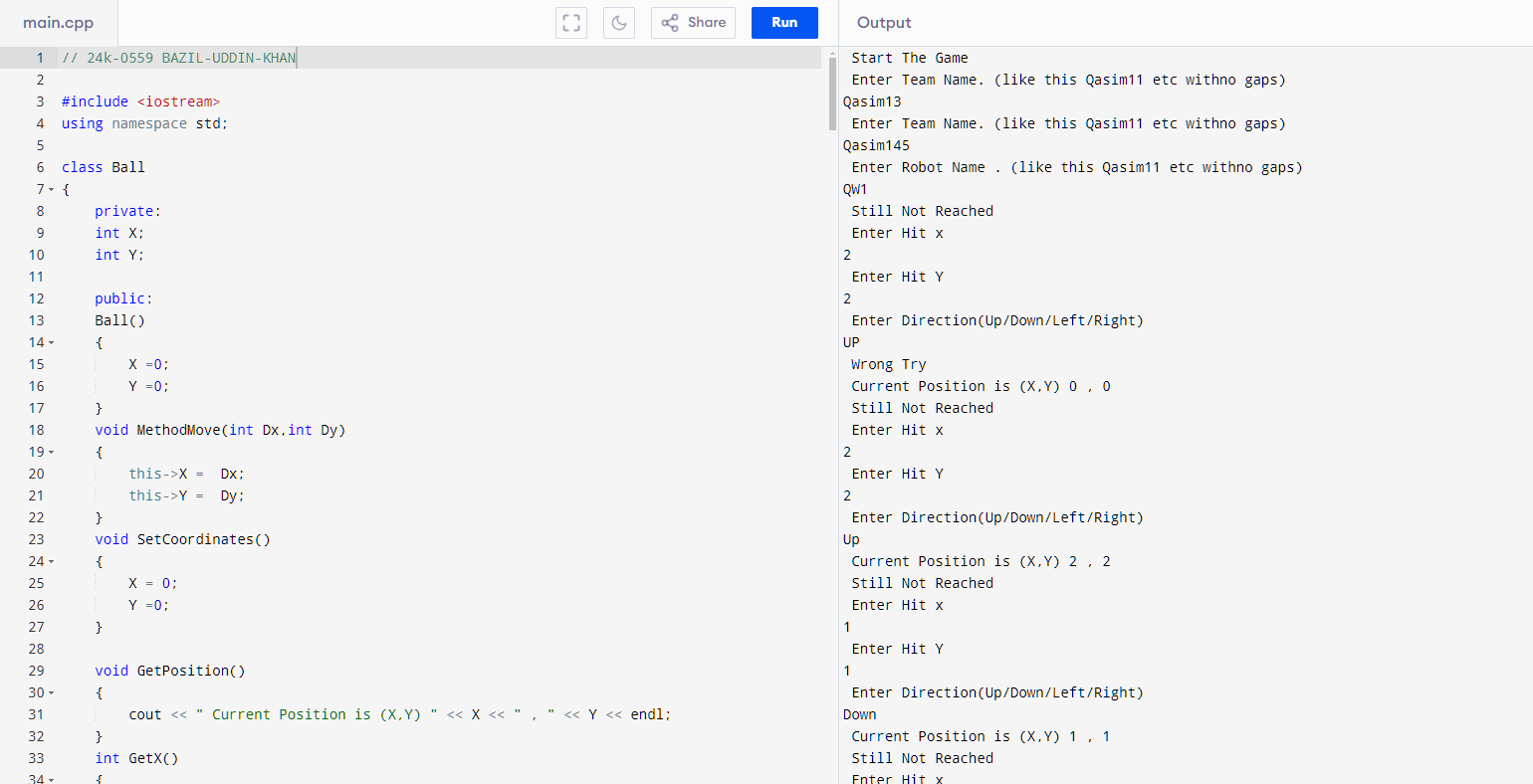
**GAME.DeclareWinner(team);**

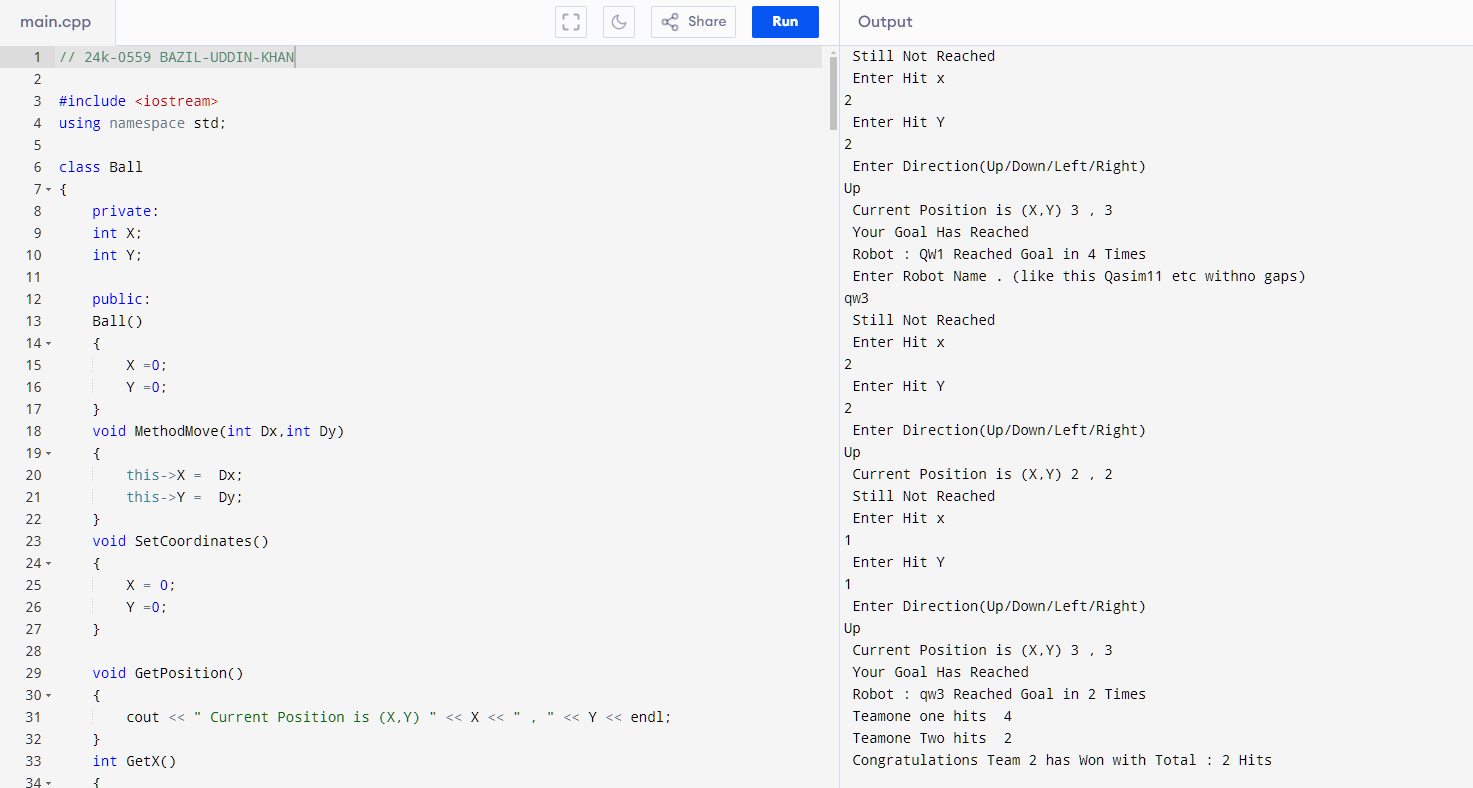
**delete [] team;**

**return 0;**

**}**

**QUESTION 2 OUTPUT:**

****

****

**QUESTION 3).**

**CODE:**

**// 24K-0559 BAZIL-UDDIN-KHAN**

**#include <iostream>**

**using namespace std;**

**class UserRegister**

**{**

**private:**

**int Age;**

**string LiscenceType;**

**string ContactNumber;**

**string UserId;**

**public:**

**UserRegister()**

**{**

**Age = 0;**

**LiscenceType = " ";**

**ContactNumber =" ";**

**UserId = " ";**

**}**

**string GetUserId()**

**{**

**return UserId;**

**}**

**void SetAge(int Age)**

**{**

**this->Age = Age;**

**}**

**void SetLiscenceType(string LiscenceType)**

**{**

**this->LiscenceType = LiscenceType;**

**}**

**void SetContactNum(string ContactNumber)**

**{**

**this->ContactNumber = ContactNumber;**

**}**

**void RegisterDeatils()**

**{**

**int Age;**

**cout << " Enter Your Age " << endl;**

**cin >> Age;**

**this->Age = Age;**

**string LiscenceType;**

**cout << " Enter Your Liscence Type(Full,Intermediate,Learners) " << endl;**

**cin >> LiscenceType;**

**this->LiscenceType = LiscenceType;**

**string ContactNumber;**

**cout << " Enter Your Contact Number " << endl;**

**cin >> ContactNumber;**

**this->ContactNumber = ContactNumber;**

**string UserId;**

**cout << " Enter Your User Id " << endl;**

**cin >> UserId;**

**this->UserId = UserId;**

**}**

**void UpdateDetails(UserRegister\* user,int j)**

**{**

**int Age;**

**cout << " Enter Your Updated Age " << endl;**

**cin >> Age;**

**user[j].SetAge(Age);**

**string LiscenceType;**

**cout << " Enter Your Updated Liscence Type " << endl;**

**cin >> LiscenceType;**

**user[j].SetLiscenceType(LiscenceType);**

**string ContactNumber;**

**cout << " Enter Your Updated Contact Number " << endl;**

**cin >> ContactNumber;**

**user[j].SetContactNum(ContactNumber);**

**}**

**void CallCapplicationStatus()**

**{**

**ApplicationStatus();**

**}**

**void StartApplication()**

**{**

**cout << " Start Your Application " << endl;**

**}**

**void EndApplication()**

**{**

**cout << " Application Ended " << endl;**

**}**

**int ApplicationStatus()**

**{**

**return 1;**

**}**

**string GetLiscenceType()**

**{**

**return LiscenceType;**

**}**

**};**

**class Vehicle**

**{**

**private:**

**string Model;**

**double RentalPrice;**

**string EligiblityStatus;**

**string LearnersVehicles[10] = {"NisanVersa","Toyotayaris","Kia","Hyundai","ToyotaCarolla","SuzukiAlto","SuzukiMehran","Honda","HondaGr","Honda360" };**

**double LearnerCarsRentalPrice[10] = {3235.56,4300,7500.67,11000,4200,9000,6700,2034,5000,3000};**

**string IntermediateVehicles[10] = {"NisanVersa","Toyotayaris","Kia","Hyundai","ToyotaCarolla","SuzukiAlto","SuzukiMehran","Honda","HondaGr","Honda360" };**

**double IntermediateCarsRentalPrice[10] = {12355.56,13000,14500.67,11000,14200,19000,67000,2034,5000,30000};**

**string FullVehicles[10] = {"HondaBrv","KiaPicanto","KiaSportage","Ford","Toyataparado","SuzukiAlto","SuzukiMehran","Honda","HondaGr","Honda360" };**

**double FullCarsRentalPrice[10] = {123555.56,130000,145000.67,110000,142000,190000,67000,203004,500000,30000};**

**void SetAttributes(string Model,double RentalPrice,string EligiblityStatus)**

**{**

**this->Model = Model;**

**this->RentalPrice = RentalPrice;**

**this->EligiblityStatus = EligiblityStatus;**

**}**

**public:**

**Vehicle()**

**{**

**Model = " ";**

**RentalPrice =0;**

**EligiblityStatus = " ";**

**}**

**Vehicle(string Model,double RentalPrice,string EligiblityStatus)**

**{**

**this->Model = Model;**

**this->RentalPrice = RentalPrice;**

**this->EligiblityStatus = EligiblityStatus;**

**}**

**void BuiltInWork(Vehicle \*\* vehicle)**

**{**

**for(int i =0; i < 10; i++)**

**{**

**vehicle[i]->SetAttributes(LearnersVehicles[i],LearnerCarsRentalPrice[i],"Learners");**

**}**

**for(int i =0; i < 10; i++)**

**{**

**vehicle[i+10]->SetAttributes(IntermediateVehicles[i],IntermediateCarsRentalPrice[i],"Intermediate");**

**}**

**for(int i =0; i < 10; i++)**

**{**

**vehicle[i+20]->SetAttributes(FullVehicles[i],FullCarsRentalPrice[i],"Full");**

**}**

**}**

**string GetEligiblityStatus()**

**{**

**return EligiblityStatus;**

**}**

**double GetRentalRate()**

**{**

**return RentalPrice;**

**}**

**void Display(string Type)**

**{**

**if(Type == "Learners")**

**{**

**cout << " Your VehicleLisence Is " << "Learners" << endl;**

**cout << " Available Care Are " << endl;**

**cout << " Cars " << " " << " Rents " << endl;**

**for(int i =0; i < 10; i++)**

**{**

**cout << " " << LearnersVehicles[i]**

**<< "\t\t\t" << LearnerCarsRentalPrice[i] << " " << endl;**

**}**

**}**

**else if(Type == "Intermediate")**

**{**

**cout << " Your VehicleLisence Is " << "Intermediate" << endl;**

**cout << " Available Care Are " << endl;**

**cout << " Cars " << " " << " Rents " << endl;**

**for(int i =0; i < 10; i++)**

**{**

**cout << " " << IntermediateVehicles[i]**

**<< "\t\t\t" << IntermediateCarsRentalPrice[i] << " " << endl;**

**}**

**}**

**else if(Type == "Full")**

**{**

**cout << " Your VehicleLisence Is " << "Full" << endl;**

**cout << " Available Care Are " << endl;**

**cout << " Cars " << " " << " Rents " << endl;**

**for(int i =0; i < 10; i++)**

**{**

**cout << " " << FullVehicles[i]**

**<< "\t\t\t" << FullCarsRentalPrice[i] << " " << endl;**

**}**

**}**

**else**

**{**

**cout << " Invalid " << endl;**

**}**

**}**

**void CheckUserRequirements(string VehicleName,UserRegister \* user,int j,Vehicle \*\* vehicle)**

**{**

**string LiscenceType = user[j].GetLiscenceType();**

**int Found = -1;**

**if(LiscenceType == "Learners")**

**{**

**for(int i =0; i < 10; i++)**

**{**

**if(vehicle[i]->GetEligiblityStatus() == LiscenceType && LearnersVehicles[i] == VehicleName)**

**{**

**cout << " Vehicle is Available and You are Competent To use it " << endl;**

**cout << " Vehicle name is " << LearnersVehicles[i] << " " << "Vehicle Rent is : " << " " << vehicle[i]->GetRentalRate() << endl;**

**Found =0;**

**break;**

**}**

**}**

**}**

**else if(LiscenceType == "Intermediate")**

**{**

**for(int i =0; i < 10; i++)**

**{**

**if(vehicle[i+10]->GetEligiblityStatus() == LiscenceType && IntermediateVehicles[i] == VehicleName)**

**{**

**cout << " Vehicle is Available and You are Competent To use it " << endl;**

**cout << " Vehicle name is " << IntermediateVehicles[i] << " " << " Vehicle Rent is : " << " " << vehicle[i]->GetRentalRate() << endl;**

**Found =0;**

**break;**

**}**

**}**

**}**

**else if(LiscenceType == "Full")**

**{**

**for(int i =0; i < 10; i++)**

**{**

**if(vehicle[i+20]->GetEligiblityStatus() == LiscenceType && FullVehicles[i] == VehicleName)**

**{**

**cout << " Vehicle is Available and You are Competent To use it " << endl;**

**cout << " Vehicle name is " << FullVehicles[i] << " " << "Vehicle Rent is : " << " " << vehicle[i+20]->GetRentalRate() << endl;**

**Found =0;**

**break;**

**}**

**}**

**}**

**if(Found == -1)**

**{**

**cout << " Sorry You Are Not Eligible For This Car " << endl;**

**}**

**}**

**~Vehicle()**

**{**

**}**

**};**

**int main()**

**{**

**cout << " Welcome to vehicle rental system " << endl;**

**UserRegister \* user = new UserRegister[1000];**

**int i=0;**

**Vehicle \*\* vehicle = new Vehicle\*[30];**

**for(int y =0; y < 30; y++ )**

**{**

**vehicle[y] = new Vehicle();**

**}**

**(vehicle[i])->BuiltInWork(vehicle);**

**user[i].StartApplication();**

**int Count =0;**

**while(1)**

**{**

**string Choice;**

**cout << " Enter Your Choice (Registerinfo/Update/RegisterVehicle/Exit/Display) ";**

**cin >> Choice;**

**if(Choice == "Registerinfo")**

**{**

**user[i].RegisterDeatils();**

**i++;**

**Count =1;**

**}**

**else if(Choice == "Update"&&Count!=0)**

**{**

**string UserId;**

**cout << " Enter User Id For Updation " << endl;**

**cin >> UserId;**

**for(int j =0; j <= i;j++)**

**{**

**if(user[j].GetUserId() == UserId)**

**{**

**user[j].UpdateDetails(user,j);**

**break;**

**}**

**}**

**}**

**else if(Choice == "RegisterVehicle"&&Count!=0)**

**{**

**string VehicleName;**

**cout << " Enter Vehicle Name You want To hire " << endl;**

**cin >> VehicleName;**

**(vehicle[i])->CheckUserRequirements(VehicleName,user,i-1,vehicle);**

**}**

**else if(Choice == "Display" &&Count!=0)**

**{**

**string Type = user[i-1].GetLiscenceType();**

**vehicle[i-1]->Display(Type);**

**}**

**else if(Choice == "Exit")**

**{**

**user[i].EndApplication();**

**break;**

**}**

**else if(Count == 0)**

**{**

**cout << " First Register "<<endl;**

**}**

**else**

**{**

**cout << " Invalid Input " << endl;**

**}**

**}**

**for(int j =0; j < 30;j++)**

**{**

**delete vehicle[j];**

**}**

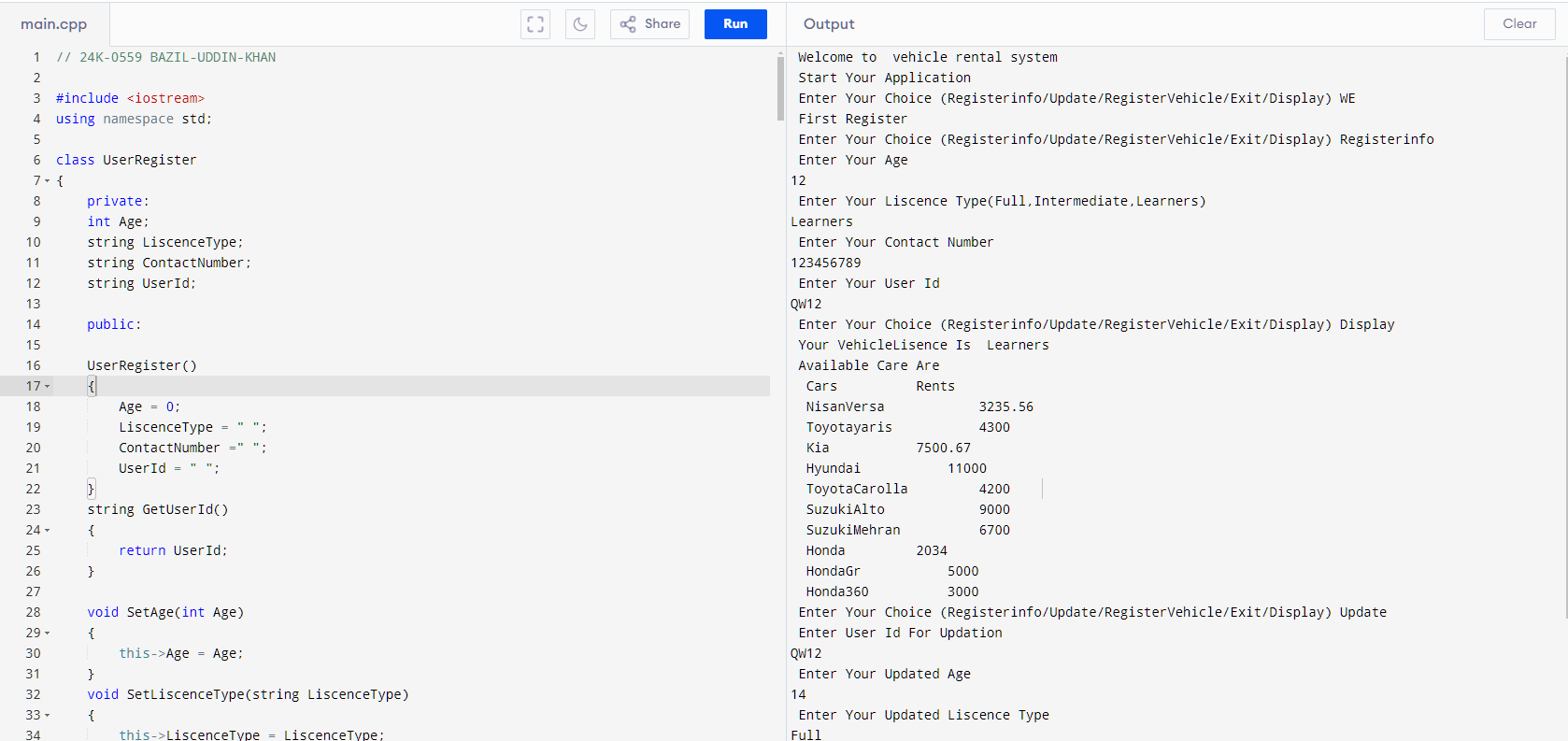
**delete [] vehicle;**

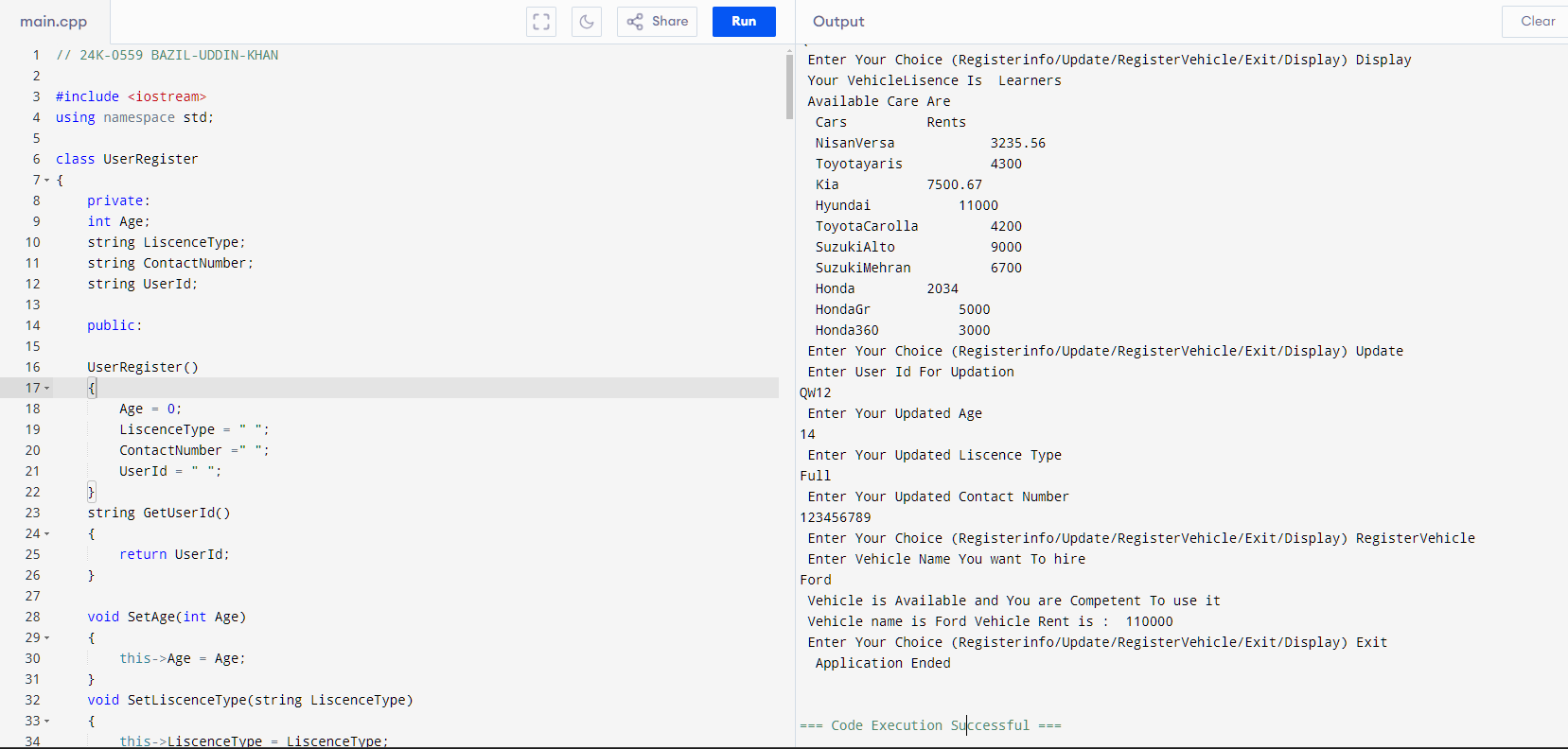
**delete [] user;**

**return 0;**

**}**

**QUESTION 3 OUTPUT :**

****

****

**QUESTION 4).**

**CODE:**

**// 24K-0559 BAZIL-UDDIN-KHAN**

**#include <iostream>**

**#include<cstring>**

**#include<string>**

**using namespace std;**

**class Routes**

**{**

**private:**

**string Routes[24] = {"Roùte2","Route3","Route3B","Route4","Route4B","Route5","Route5B","Route6","Route6B","Route7","Route8","Route9","Route17","Route18","Route20","Route22","Route23","Route23B","Route24","Route24B","Route25","Route25B","Route26","Route26B"};**

**string RoutesPickUp[24] = {"Saffora","SindhBaloach","KamranChorangi","Jamalipull","Maskan","PowerHouse","DolminMall","bardadari","Golbarq","Donisel","clifton","Bahria","Dha","UniversityRoad","Korangi","lyaripull","Motimehal","Sheefaisal","Balochistansajji","Kdm","Mehran","Lalqila","Tariqroad","Balochistanx"};**

**public:**

**int CheckRoute(string Route)**

**{**

**for(int i=0; i < 24; i++)**

**{**

**if(Routes[i] == Route)**

**{**

**return 1;**

**}**

**}**

**cout << " Not Found " << endl;**

**return -1;**

**}**

**string GetRouteName(string RouteNumber)**

**{**

**for(int i =0; i < 24;i++)**

**{**

**if(Routes[i] == RouteNumber)**

**{**

**string val = RoutesPickUp[i];**

**return(val);**

**}**

**}**

**return " ";**

**}**

**};**

**class PaymentFee**

**{**

**private:**

**double Fees;**

**string CardStatus;**

**public:**

**PaymentFee()**

**{**

**Fees =0;**

**CardStatus = "NotPaid";**

**}**

**PaymentFee(double Fee)**

**{**

**this->Fees = Fee;**

**}**

**int PaymentFees(double fees)**

**{**

**if(Fees <= fees)**

**{**

**this->Fees = Fees;**

**this->CardStatus = "Paid";**

**cout << " Payment Susssfully made " << endl;**

**return 1;**

**}**

**else**

**{**

**cout << " Payment Not Recieved " << endl;**

**return 0;**

**}**

**}**

**string GetStatus()**

**{**

**return CardStatus;**

**}**

**};**

**class Student**

**{**

**private:**

**string StudentId;**

**string StudentName;**

**string StudentContactNumber;**

**string BatchNo;**

**string SemesterYear;**

**string RouteNumber;**

**string StopName;**

**static int TotalStudents;**

**string CardNumber;**

**Routes route;**

**public:**

**Student()**

**{**

**StudentId = " ";**

**StudentName = " ";**

**StudentContactNumber = " ";**

**BatchNo = " ";**

**SemesterYear = " ";**

**RouteNumber = " ";**

**StopName = " ";**

**CardNumber =" ";**

**}**

**Student(string StudentId,string StudentName,string StudentContactNumber, string BatchNo,string SemesterYear,string RouteNumber,string StopName,string CardNumber)**

**{**

**this->StudentId = StudentId;**

**this->StudentName = StudentName;**

**this->StudentContactNumber = StudentContactNumber;**

**this->BatchNo = BatchNo;**

**this->SemesterYear = SemesterYear;**

**this->RouteNumber = RouteNumber;**

**this->StopName = StopName;**

**this->CardNumber = CardNumber;**

**}**

**void SetAttributes(string StudentId,string StudentName,string StudentContactNumber, string BatchNo,string SemesterYear,string RouteNumber,string StopName,string CardNumber)**

**{**

**this->StudentId = StudentId;**

**this->StudentName = StudentName;**

**this->StudentContactNumber = StudentContactNumber;**

**this->BatchNo = BatchNo;**

**this->SemesterYear = SemesterYear;**

**this->RouteNumber = RouteNumber;**

**this->StopName = StopName;**

**this->CardNumber = CardNumber;**

**}**

**void StudentRegisteration()**

**{**

**string StudentId;**

**cout << " Enter Your Fast Id like this(21k-0678) " << endl;**

**cin >> StudentId;**

**string StudentName;**

**cout << " Enter Your Student Name " << endl;**

**cin.ignore();**

**getline(cin,StudentName);**

**string StudentContactNumber;**

**cout << " Enter Student Contact Number " << endl;**

**cin >> StudentContactNumber;**

**string BatchNo;**

**cout << " Enter Batch No like 2022 " << endl;**

**cin >> BatchNo;**

**string SemesterYear;**

**cout << " Enter Semester(Fall/Spring)Year like Fall2024 ";**

**cin >> SemesterYear;**

**string RouteNumber;**

**cout << " Enter Route Number. Note Every Route Number start with (RouteNo) No represents The number of route like Route1 ,Route24 etc " << endl;**

**cin >> RouteNumber;**

**PaymentFee payment(21200);**

**int Routenumber = route.CheckRoute(RouteNumber);**

**if(Routenumber != -1)**

**{**

**this->RouteNumber = RouteNumber;**

**string RouteName = route.GetRouteName(RouteNumber);**

**if(RouteName != " " && payment.GetStatus() != "NotPaid")**

**{**

**this->StopName = RouteName;**

**cout << " Succesfully Registered " << endl;**

**CardNumber = ("0"+ to\_string(TotalStudents));**

**SetAttributes(StudentId,StudentName,StudentContactNumber,BatchNo,SemesterYear,RouteNumber,RouteName,CardNumber);**

**TotalStudents++;**

**}**

**else**

**{**

**cout << " Failed To Register " << endl;**

**}**

**}**

**else**

**{**

**cout << " Route Not Found " << endl;**

**}**

**}**

**string GetContactNumber() const**

**{**

**return StudentContactNumber;**

**}**

**string GetName() const**

**{**

**return StudentName;**

**}**

**static int GetTotalStudents()**

**{**

**return TotalStudents;**

**}**

**string GetUserId() const**

**{**

**return StudentId;**

**}**

**};**

**int Student :: TotalStudents =0;**

**class Attendance**

**{**

**private:**

**int TotalAttendance;**

**Student \* student = nullptr;**

**int \* Attendancechecker;**

**public:**

**Attendance(int Totalstudents,Student \* students)**

**{**

**Attendancechecker = new int [Totalstudents];**

**TotalAttendance =0;**

**student = students;**

**for(int i =0; i < Totalstudents;i++)**

**{**

**Attendancechecker[i] =0;**

**}**

**}**

**void SetCard(string UserId,int Totalstudents)**

**{**

**PaymentFee card;**

**string status = card.GetStatus();**

**for(int j =0; j < Totalstudents;j++)**

**{**

**if(student[j].GetUserId() == UserId)**

**{**

**TotalAttendance++;**

**Attendancechecker[j] = TotalAttendance;**

**}**

**}**

**}**

**void DisplayAttendance(int Totalstudents)**

**{**

**for(int i =0; i < Totalstudents;i++)**

**{**

**cout << " Attendance is of " << i+1 << " Student is " << endl;**

**cout << Attendancechecker[i] << endl;**

**cout << " Name is " << student[i].GetName() << endl;**

**cout << " Id is " << student[i].GetUserId()<<endl;**

**cout << " Contact Number is " << student[i].GetContactNumber() << endl;**

**}**

**}**

**~Attendance()**

**{**

**cout << " Memory Released " << endl;**

**delete [] Attendancechecker;**

**}**

**};**

**int main()**

**{**

**cout << " Welcome To Fast Bus Transportation System " << endl;**

**cout << endl;**

**int TotalStudents;**

**cout << " Enter Total Students who want to make there Card " << endl;**

**cin >> TotalStudents;**

**Student \* student = new Student[TotalStudents];**

**for(int i =0; i < TotalStudents;i++)**

**{**

**student[i].StudentRegisteration();**

**}**

**string userid;**

**cout << " Enter User id whose Attendance is to be noted " << endl;**

**cin >> userid;**

**Attendance attendance(TotalStudents,student);**

**attendance.SetCard(userid,TotalStudents);**

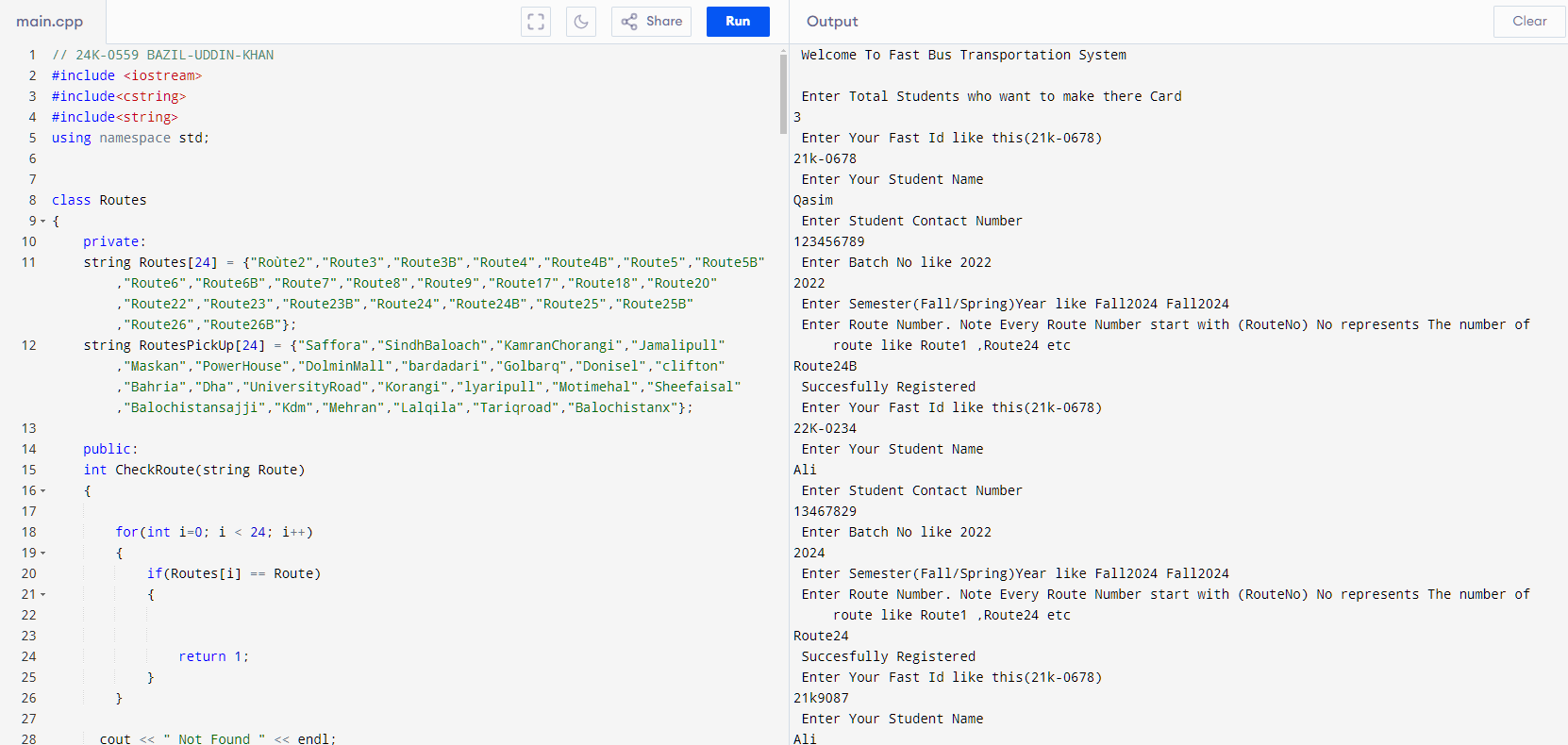
**attendance.DisplayAttendance(TotalStudents);**

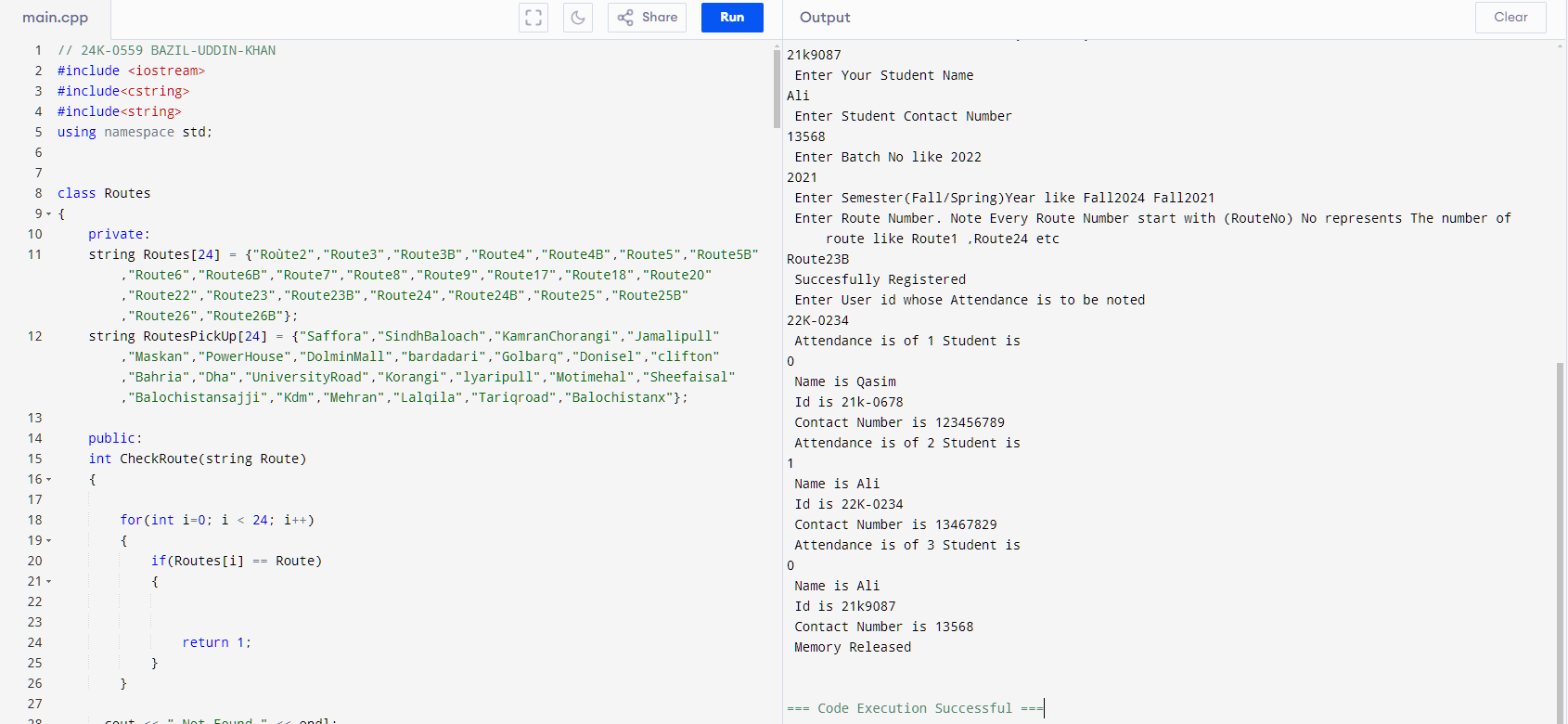
**delete [] student;**

**return 0;**

**}**

**QUESTION 4 OUTPUT:**

****

****