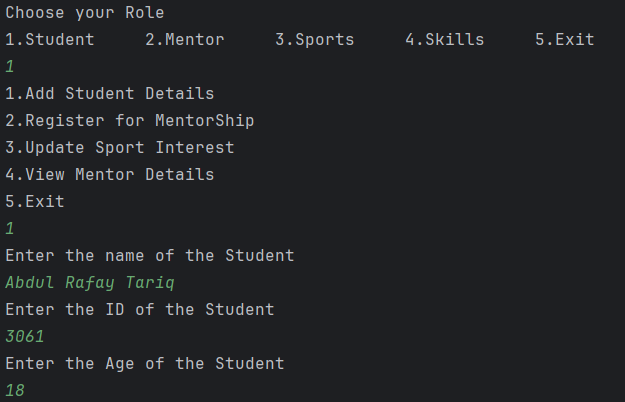
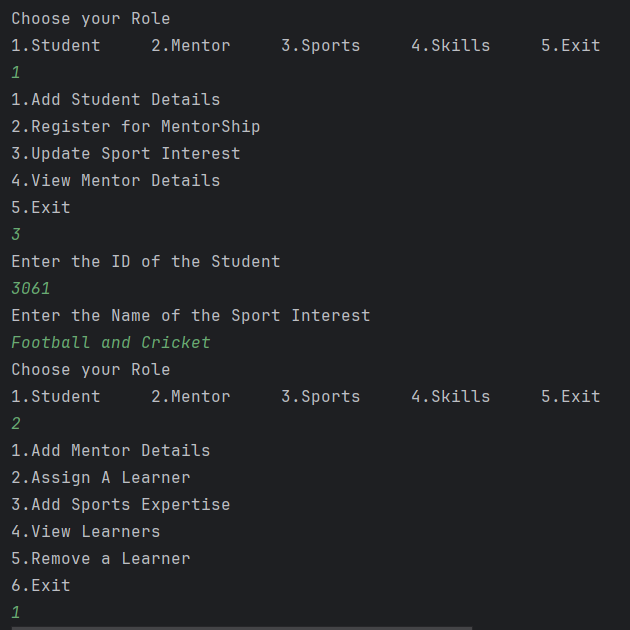
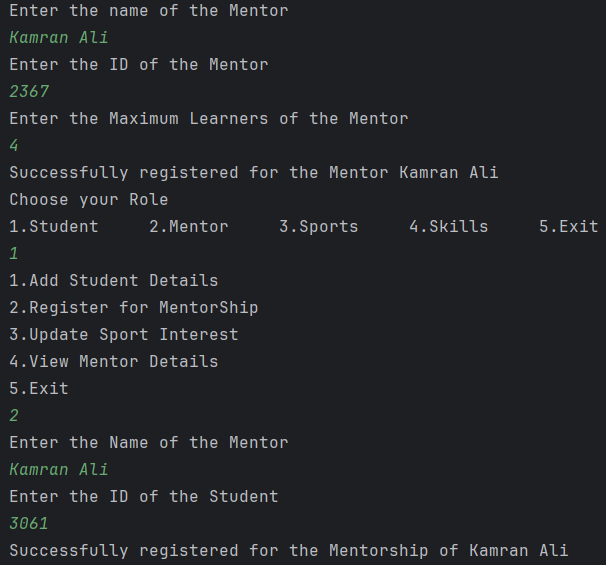
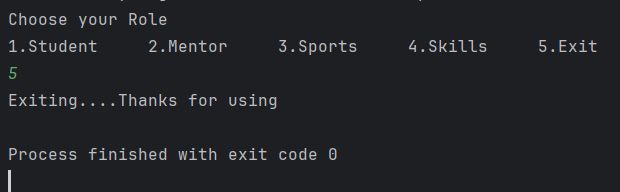
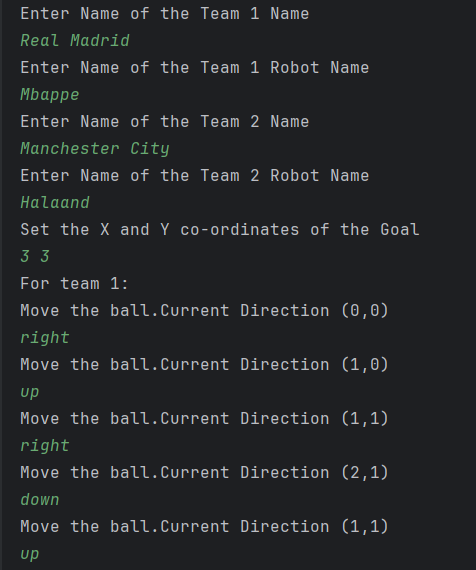
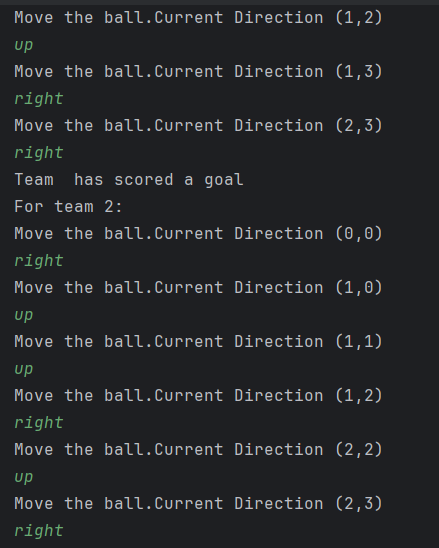
***(All Questions Code is Displayed after all Outputs)***

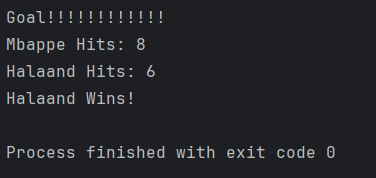
***OUTPUT:***

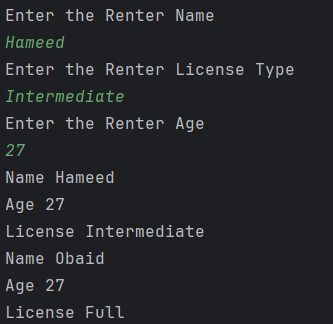
**Question 1:**

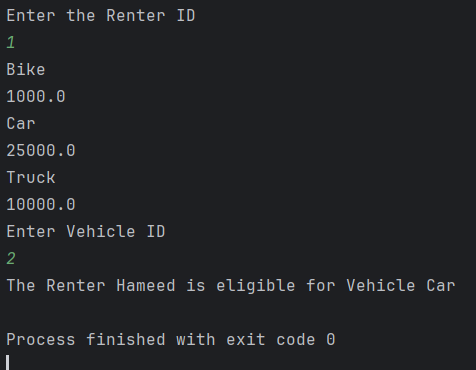




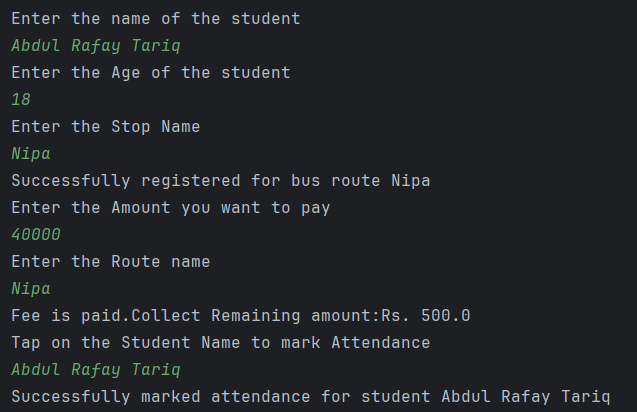
Question 2:

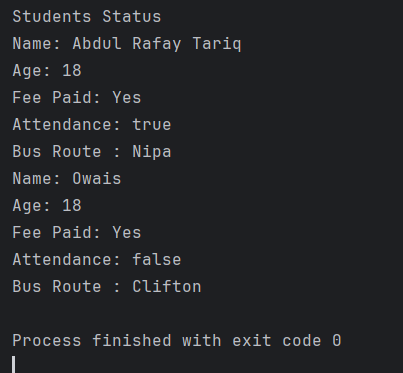


Question 3:



Question 4:





***Class Diagram:***

***+-------------------------------+***

***| Student |***

***+-------------------------------+***

***| - studentName: String |***

***| - studentAge: int |***

***| - feePaid: boolean |***

***| - attendance: boolean |***

***+-------------------------------+***

***| + Student(String, int) |***

***| + getStudentName(): String |***

***| + payTransportFee(): void |***

***| + studentStatus(): void |***

***+-------------------------------+***

***▲***

***│***

***│***

***│***

***+-------------------------------+***

***| BusManager |***

***+-------------------------------+***

***| - managerName: String |***

***| - busFee: long |***

***| - maxStudents: static int |***

***| - busRoutes: ArrayList<BusRoute> |***

***| - students: ArrayList<Student> |***

***+-------------------------------+***

***| + BusManager() |***

***| + stop(BusRoute): void |***

***| + addStudent(Student): void |***

***| + registerBusRoute(String, String): void |***

***| + markAttendance(String): void |***

***| + payFee(double, String, String): void |***

***+-------------------------------+***

***▲***

***│***

***│***

***│***

***+-------------------------------+***

***| BusRoute |***

***+-------------------------------+***

***| - routeName: String |***

***| - totalStudents: int |***

***+-------------------------------+***

***| + addBusRoute(String, int): void |***

***| + displayBusRoute(): void |***

***| + getBusRoute(): String |***

***+-------------------------------+***

***▲***

***│***

***│***

***│***

***+------------------------------------------+***

***| TransportationManagementSystem (Main) |***

***+------------------------------------------+***

***| + main(String[] args): void |***

***+------------------------------------------+***

***Code:***

**Question 1:**

**Mentor Class:**

package Q1;  
  
import java.util.ArrayList;  
  
public class Mentor {  
 String mentorName;  
 int mentorID;  
 ArrayList<Student>assignedLearners;  
 int maxLearners;  
 ArrayList<Sport>sportExpertise;  
 public Mentor() {  
 this.mentorName="Unknown";  
 this.mentorID=-1;  
 this.assignedLearners=new ArrayList<>();  
 this.sportExpertise=new ArrayList<>();  
 this.maxLearners=0;  
 }  
 public Mentor(String mentorName,int mentorID,int maxLearners) {  
 this.mentorName = mentorName;  
 this.mentorID = mentorID;  
 this.maxLearners = maxLearners;  
 this.assignedLearners = new ArrayList<>();  
 }  
 public void assignLearners(Student student) {  
 if (assignedLearners.size()<maxLearners) {  
 this.assignedLearners.add(student);  
 }  
 }  
 public void addSportExpertise(Sport sport){  
 this.sportExpertise.add(sport);  
 }  
 public void removeLearners(Student student) {  
 this.assignedLearners.remove(student);  
  
 System.*out*.println("Removed "+student.studentID);  
 }  
 public void provideGuidance(){  
 System.*out*.println("GUIDANCE :\n");  
 }  
 public void viewAssignedLearners() {  
 System.*out*.println("Assigned Learners: " + this.assignedLearners);  
 }  
 public void viewMentorDetails(){  
 System.*out*.println("Mentor Details");  
 System.*out*.println("Name: " + this.mentorName);  
 System.*out*.println("ID: " + this.mentorID);  
 System.*out*.println("Sports Expertise: " + this.sportExpertise);  
 }  
  
}

**Student Class**:

package Q1;  
  
import java.util.ArrayList;  
  
public class Student {  
 String studentName;  
 int studentID;  
 int studentAge;  
 ArrayList<String>sportsInterests;  
 Mentor mentorAssigned;  
 public Student(String studentName,int studentID,int studentAge) {  
 this.studentName = studentName;  
 this.studentID = studentID;  
 this.studentAge = studentAge;  
 sportsInterests = new ArrayList<>();  
 }  
  
 public Student() {  
 this.studentName = "";  
 this.studentID = 0;  
 this.studentAge = 0;  
 sportsInterests = new ArrayList<>();  
 }  
  
 public void registerForMentorShip(Mentor mentor) {  
 this.mentorAssigned = mentor;  
 }  
 public void updateSportsInterest(String sportsInterest) {  
 sportsInterests.add(sportsInterest);  
 }  
 public void viewMentorDetails()  
 {  
 System.*out*.println("Name: " + mentorAssigned.mentorName);  
 System.*out*.println("ID: " + mentorAssigned.mentorID);  
 System.*out*.println("Max Learners: "+mentorAssigned.maxLearners);  
 }  
  
  
  
}

**Sport Class:**

package Q1;  
  
import java.util.ArrayList;  
  
public class Sport {  
 String sportName;  
 String sportDescription;  
 int sportID;  
 ArrayList<Skill> requiredSkills;  
 public Sport(String sportName,int sportID,String sportDescription){  
 this.sportName = sportName;  
 this.sportID = sportID;  
 this.sportDescription = sportDescription;  
 this.requiredSkills = new ArrayList<>();  
 }  
 public void addRequiredSkill(Skill skill){  
 requiredSkills.add(skill);  
 }  
 public void removeRequiredSkill(int id){  
 for (Skill skill:requiredSkills){  
 if (skill.skillID==id){  
 requiredSkills.remove(skill);  
 }  
 }  
 }  
public void displaySportDetails(){  
 System.*out*.println("Name: " + sportName);  
 System.*out*.println("ID: " + sportID);  
 System.*out*.println("Description: " + sportDescription);  
 System.*out*.println("Required Skills: " + requiredSkills);  
}  
}

**Skill Class:**

**package Q1;**public class Skill {  
 String skillName;  
 int skillID;  
 String skillDescription;  
 public Skill(String skillName, int skillID, String skillDescription) {  
 super();  
 this.skillName = skillName;  
 this.skillID = skillID;  
 this.skillDescription = skillDescription;  
 }  
 public void showSkillDescription(){  
 System.*out*.println("Skill Name: " + skillName);  
 System.*out*.println("Skill ID: " + skillID);  
 System.*out*.println("Skill Description: " + skillDescription);  
 }  
 public void updateSkillDescription(String skillDescription){  
 this.skillDescription = skillDescription;  
 }  
}

**Main Class:**

package Q1;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class SportsManagementSystem {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 ArrayList<Student> students = new ArrayList<Student>();  
 ArrayList<Mentor> mentors = new ArrayList<>();  
 ArrayList<Sport> sports = new ArrayList<>();  
 ArrayList<Skill> skills = new ArrayList<>();  
 int roleChoice;  
 int studentFound = 0;  
 int mentorFound = 0;  
 do {  
 System.*out*.println("Choose your Role\n1.Student 2.Mentor 3.Sports 4.Skills 5.Exit");  
 roleChoice = scan.nextInt();  
 switch (roleChoice) {  
 case 1:  
 scan.nextLine();  
 System.*out*.println("1.Add Student Details\n2.Register for MentorShip\n3.Update Sport Interest\n4.View Mentor Details\n5.Exit");  
 int choice;  
 choice = scan.nextInt();  
 switch (choice) {  
 case 1:  
 scan.nextLine();  
 System.*out*.println("Enter the name of the Student");  
 String studentName = scan.nextLine();  
 System.*out*.println("Enter the ID of the Student");  
 int studentID = scan.nextInt();  
 System.*out*.println("Enter the Age of the Student");  
 int age = scan.nextInt();  
 scan.nextLine();  
 students.add(new Student(studentName, studentID, age));  
 break;  
 case 2:scan.nextLine();  
 System.*out*.println("Enter the Name of the Mentor");  
 String mentorName = scan.nextLine();  
 System.*out*.println("Enter the ID of the Student");  
 int sID = scan.nextInt();  
 scan.nextLine();  
 for (Mentor mentor : mentors) {  
 if (mentor.mentorName.equals(mentorName)) {  
 mentorFound = 1;  
 for (Student student : students) {  
 if (student.studentID == sID) {  
 studentFound = 1;  
 student.registerForMentorShip(mentor);  
 System.*out*.println("Successfully registered for the Mentorship of " + mentor.mentorName);  
 break;  
 }  
 }  
 }  
 }  
 if (studentFound == 0 || mentorFound == 0) {  
 System.*out*.println("Invalid Mentor or Student details entered");  
 }  
 break;  
 case 3:  
 scan.nextLine();  
 System.*out*.println("Enter the ID of the Student");  
 int s\_id = scan.nextInt();  
 scan.nextLine();  
 studentFound = 0;  
 for (Student student : students) {  
 if (student.studentID == s\_id) {  
 studentFound = 1;  
 System.*out*.println("Enter the Name of the Sport Interest");  
 String sportInterest = scan.nextLine();  
 student.updateSportsInterest(sportInterest);  
 }  
 }  
 if (studentFound==0) {  
 System.*out*.println("Student not found");  
 }  
 break;  
 case 4:  
 if (mentors.isEmpty()) {  
 System.*out*.println("No Mentors found");  
 } else {  
 for (Mentor mentor : mentors) {  
 mentor.viewMentorDetails();  
 }  
 }  
 break;  
 case 5:  
 System.*out*.println("Returning back to the Role Menu");  
 break;  
 }  
 break;  
 case 2:  
 System.*out*.println("1.Add Mentor Details\n2.Assign A Learner\n3.Add Sports Expertise\n4.View Learners\n5.Remove a Learner\n6.Exit");  
 int choice2 = scan.nextInt();  
 scan.nextLine();  
 switch (choice2) {  
 case 1:  
 System.*out*.println("Enter the name of the Mentor");  
 String mentorName = scan.nextLine();  
 System.*out*.println("Enter the ID of the Mentor");  
 int mentorID = scan.nextInt();  
 System.*out*.println("Enter the Maximum Learners of the Mentor");  
 int maxLearners = scan.nextInt();  
 mentors.add(new Mentor(mentorName, mentorID, maxLearners));  
 System.*out*.println("Successfully registered for the Mentor " + mentorName);  
 break;  
 case 2:  
 System.*out*.println("Enter the ID of the Mentor");  
 int m\_id = scan.nextInt();  
 System.*out*.println("Enter the Name of the Learner");  
 String learnerName = scan.nextLine();  
 System.*out*.println("Enter the ID of the Learner");  
 int learnerID = scan.nextInt();  
 System.*out*.println("Enter the age of the Student");  
 int LearnerAge = scan.nextInt();  
 mentorFound = 0;  
 for (Mentor mentor : mentors) {  
 if (mentor.mentorID == m\_id) {  
 mentorFound = 1;  
 mentor.assignLearners(new Student(learnerName, learnerID, LearnerAge));  
 System.*out*.println("Successfully registered for the Mentor " + mentor.mentorName);  
 break;  
 }  
 }  
 if (mentorFound == 0) {  
 System.*out*.println("The Mentor is not found");  
 }  
 break;  
 case 3:  
 System.*out*.println("Enter the ID of the Mentor");  
 int men\_id = scan.nextInt();  
 scan.nextLine();  
 for (Mentor mentor : mentors) {  
 if (mentor.mentorID == men\_id) {  
 System.*out*.println("Enter the name of the Sport");  
 String sport = scan.nextLine();  
 System.*out*.println("Enter the ID of the Sport");  
 int sportID = scan.nextInt();  
 System.*out*.println("Enter the Description of the Sports");  
 String sportDescription = scan.nextLine();  
 mentor.addSportExpertise(new Sport(sport, sportID, sportDescription));  
 System.*out*.println("The Sports Expertise has been Added Successfully");  
 }  
 }  
 break;  
 case 4:  
 System.*out*.println("Enter the Mentor ID");  
 int id = scan.nextInt();  
 scan.nextLine();  
 mentorFound = 0;  
 for (Mentor mentor : mentors) {  
 if (mentor.mentorID == id) {  
 mentorFound = 1;  
 mentor.viewAssignedLearners();  
 }  
 }  
 if (mentorFound == 0) {  
 System.*out*.println("The Mentor is not found");  
 }  
 break;  
 case 5:  
 System.*out*.println("Enter the ID of the Mentor");  
 int mentor\_id = scan.nextInt();  
 scan.nextLine();  
 studentFound = 0;  
 mentorFound = 0;  
 for (Mentor mentor : mentors) {  
 if (mentor.mentorID == mentor\_id) {  
 mentorFound = 1;  
 System.*out*.println("Enter the Name of the Learner");  
 int s\_id = scan.nextInt();  
 for (Student student : students) {  
 if (student.studentID == s\_id) {  
 studentFound = 1;  
 mentor.removeLearners(student);  
 }  
 }  
 }  
 }  
 if (studentFound == 0 || mentorFound == 0) {  
 System.*out*.println("The Mentor or Student is not found");  
 }  
 break;  
 case 6:  
 System.*out*.println("Exiting to the Role Menu.....");  
 }  
 break;  
 case 3:  
 scan.nextLine();  
 System.*out*.println("1.Add Sports Details\n2.Add Required Skills\n3.Remove Required Skills\n4.Exit");  
 int choice3 = scan.nextInt();  
 switch (choice3) {  
 case 1:  
 System.*out*.println("Enter the Sport Name");  
 String sportName = scan.nextLine();  
 System.*out*.println("Enter the ID of the Sport");  
 int sportID = scan.nextInt();  
 scan.nextLine();  
 System.*out*.println("Enter the Description of the Sport");  
 String sportDescription = scan.nextLine();  
 sports.add(new Sport(sportName, sportID, sportDescription));  
 System.*out*.println("The Sports" + sportName + "has been Added Successfully");  
 break;  
 case 2:  
 System.*out*.println("Enter the ID of the Sport");  
 int s\_id=scan.nextInt();  
 for (Sport sport : sports) {  
 if (sport.sportID==s\_id) {  
 System.*out*.println("Enter the Required Skill Name");  
 String requiredSkill = scan.nextLine();  
 System.*out*.println("Enter the Skill ID of the Sport");  
 int skillID = scan.nextInt();  
 scan.nextLine();  
 System.*out*.println("Enter the Description of the Skill");  
 String skillDescription=scan.nextLine();  
 sport.addRequiredSkill(new Skill(requiredSkill,skillID,skillDescription));  
 break;  
 }  
 }  
 break;  
 case 3:System.*out*.println("Enter the ID of the Sport");  
 int sport\_id=scan.nextInt();  
 for (Sport sport : sports) {  
 if (sport.sportID==sport\_id) {  
 sports.remove(sport);  
 break;  
 }  
 }  
 break;  
 case 4:System.*out*.println("Exiting to the Role Menu.....");  
 break;  
 }  
 break;  
 case 4:scan.nextLine();  
 System.*out*.println("1.Add Skill Details\n2.Update Skill Description\n3.Show Skill Details\n4.Exit");  
 int choice4 = scan.nextInt();  
 scan.nextLine();  
 switch (choice4) {  
 case 1:System.*out*.println("Enter the Skill Name");  
 String skillName = scan.nextLine();  
 System.*out*.println("Enter the ID of the Skill");  
 int skillID = scan.nextInt();  
 scan.nextLine();  
 System.*out*.println("Enter the Description of the Skill");  
 String skillDescription=scan.nextLine();  
 skills.add(new Skill(skillName, skillID, skillDescription));  
 System.*out*.println("The Skills" + skillName + "has been Added Successfully");  
 break;  
 case 2:System.*out*.println("Enter the ID of the Skill");  
 int skill\_id=scan.nextInt();  
 for (Skill skill : skills) {  
 if (skill.skillID==skill\_id) {  
 System.*out*.println("Enter the Updated Skill Description");  
 String updatedSkillDescription=scan.nextLine();  
 skill.updateSkillDescription(updatedSkillDescription);  
 System.*out*.println("THe skill description has been updated Successfully");  
 break;  
 }  
 }  
 break;  
 case 3:System.*out*.println("Enter the ID of the Skill");  
 int s\_id=scan.nextInt();  
 for (Skill skill : skills) {  
 if (skill.skillID==s\_id) {  
 skill.showSkillDescription();  
 }  
 }  
 break;  
 case 4:System.*out*.println("Returning back to the Choice Menu");  
 break;  
 }  
 case 5:scan.nextLine();  
 System.*out*.println("Exiting....Thanks for using");  
 }  
 }while(roleChoice!=5);  
 }  
}

Question 2:

package Q2;  
  
public class Ball {  
 private int x,y;  
 public Ball(){  
 this.x=0;  
 this.y=0;  
 }  
 public Ball(int x, int y){  
 this.x=x;  
 this.y=y;  
 }  
 public int getX() {  
 return this.x;  
 }  
 public int getY() {  
 return this.y;  
 }  
 public void move(int dx,int dy){  
 this.x+=dx;  
 this.y+=dy;  
 }  
 public Ball getPosition(){  
 return new Ball(this.x,this.y);  
 }  
}

package Q2;  
  
public class Robot {  
 private String robotName;  
 private int noOfHits;  
  
 public Robot(String robotName) {  
 this.robotName = robotName;  
 this.noOfHits = 0;  
 }  
  
 public void hitBall(Ball ball, final String direction) {  
 this.noOfHits++;  
 switch (direction.toLowerCase()) {  
 case "up":  
 ball.move(0, 1);  
 break;  
 case "down":  
 ball.move(-1, 0);  
 break;  
 case "left":  
 ball.move(-1, 1);  
 break;  
 case "right":  
 ball.move(1, 0);  
 break;  
 default:  
 System.*out*.println("Enter a valid direction");  
 }  
 }  
 public int getNoofHits() {  
 return this.noOfHits;  
 }  
 public String getRobotName() {  
 return this.robotName;  
 }  
}

package Q2;  
  
import java.util.Scanner;  
  
public class Game {  
 Robot robot1;  
 Robot robot2;  
 Ball ball;  
 GoalPosition goal;  
 public Game(Robot robot1, Robot robot2, Ball ball, int x,int y){  
 this.robot1 = robot1;  
 this.robot2 = robot2;  
 this.ball = ball;  
 this.goal = new GoalPosition(x,y);  
 }  
 public void startGame(){  
 System.*out*.println("For team 1:");  
 playGame(robot1);  
resetBall();  
System.*out*.println("For team 2:");  
playGame(robot2);  
declareWinner();  
 }  
 public void playGame(Robot robot1){  
 Scanner scan=new Scanner(System.*in*);  
 while (!goal.isGoalReached(this.ball)){  
 System.*out*.println("Move the ball.Current Direction ("+this.ball.getX()+","+this.ball.getY()+")");  
 String direction = scan.nextLine();  
 robot1.hitBall(this.ball,direction);  
 }  
  
 System.*out*.println("Goal!!!!!!!!!!!!");  
}  
 public void declareWinner(){  
 int team1Hits=this.robot1.getNoofHits();  
 int team2Hits=this.robot2.getNoofHits();  
 System.*out*.println(robot1.getRobotName()+" Hits: "+team1Hits);  
 System.*out*.println( robot2.getRobotName()+" Hits: "+team2Hits);  
 if(team1Hits>team2Hits){  
 System.*out*.println(robot2.getRobotName()+" Wins!");  
 } else if (team2Hits>team1Hits) {  
 System.*out*.println(robot1.getRobotName()+"Wins! ");  
 }else{  
 System.*out*.println("It's a tie!");  
 }  
 }  
 public void resetBall(){  
 this.ball = new Ball();  
 }  
}

package Q2;  
  
public class GoalPosition {  
 int x;  
 int y;  
 public GoalPosition(){  
 this.x=0;  
 this.y=0;  
 }  
 public GoalPosition(int x,int y){  
 this.x=x;  
 this.y=y;  
 }  
 public boolean isGoalReached(Ball ball){  
if (this.x== ball.getX() && this.y== ball.getY()){  
 return true;  
}else{  
 return false;  
}  
 }  
}

package Q2;  
  
public class Team {  
 String teamName;  
 Robot robot;  
 Team(String teamName, String robotName) {  
 this.teamName = teamName;  
 this.robot = new Robot(robotName);  
 }  
}

package Q2;  
  
import java.util.Scanner;  
  
public class Match {  
 public static void main(String[] args) {  
 Scanner scan=new Scanner(System.*in*);  
 System.*out*.println("Enter Name of the Team 1 Name");  
 String nameTeam1=scan.nextLine();  
 System.*out*.println("Enter Name of the Team 1 Robot Name");  
 String nameRobot1=scan.nextLine();  
 System.*out*.println("Enter Name of the Team 2 Name");  
 String nameTeam2=scan.nextLine();  
 System.*out*.println("Enter Name of the Team 2 Robot Name");  
 String nameRobot2=scan.nextLine();  
 Team team1 = new Team(nameTeam1,nameRobot1);  
 Team team2 = new Team(nameTeam2,nameRobot2);  
 System.*out*.println("Set the X and Y co-ordinates of the Goal");  
 int x=scan.nextInt();  
 int y=scan.nextInt();  
 scan.nextLine();  
 GoalPosition goalPosition = new GoalPosition(x,y);  
 Ball ball=new Ball();  
 Game game=new Game(team1.robot,team2.robot,ball,x,y);  
game.startGame();  
 }  
}

***Question 3:***

package Q3;  
  
public class Renter {  
 String renterName;  
 int renterAge;  
 String renterLicense;  
 public Renter(){  
 this.renterName="Unknown";  
 this.renterAge=0;  
 this.renterLicense="UnAssigned";  
 }  
 public Renter(String renterName, int renterAge, String renterLicense) {  
 this.renterName = renterName;  
 this.renterAge = renterAge;  
 this.renterLicense = renterLicense;  
 }  
 public void displayRenterDetails(){  
 System.*out*.println("Name "+this.renterName);  
 System.*out*.println("Age "+this.renterAge);  
 System.*out*.println("License "+this.renterLicense);  
 }  
}

package Q3;  
  
import java.util.ArrayList;  
  
public class Vehicle {  
 String vehicleModel;  
 double pricePerDay;  
 ArrayList<String>eligibleVehicles;  
  
 public Vehicle(String vehicleModel, double pricePerDay,String vehicleLicenseType) {  
 this.vehicleModel=vehicleModel;  
 this.pricePerDay=pricePerDay;  
 this.eligibleVehicles=new ArrayList<>();  
 eligibleVehicles.add(vehicleLicenseType);  
 }  
  
 public boolean isEligible(String renterLicense) {  
 return eligibleVehicles.contains(renterLicense);  
 }  
 public void displayVehicleDetails() {  
 System.*out*.println(this.vehicleModel);  
 System.*out*.println(this.pricePerDay);  
 }  
}

package Q3;   
  
import javax.sound.midi.MidiFileFormat;  
import java.util.ArrayList;  
  
public class RentalSystem {  
 ArrayList<Renter>renters=new ArrayList<>();  
 ArrayList<Vehicle>vehicles=new ArrayList<>();  
 public void registerRenter(Renter renter) {  
 renters.add(renter);  
 }  
 public void registerVehicle(Vehicle vehicle) {  
 vehicles.add(vehicle);  
 }  
 public void displayRenters() {  
 for (Renter r : renters) {  
 r.displayRenterDetails();  
 }  
 }  
 public void displayVehicles() {  
 for (Vehicle v : vehicles) {  
 v.displayVehicleDetails();  
 }  
 }  
  
 public void rentVehicle(int renterId, int vehicleId) {  
 if (renterId > 0 && renterId <= renters.size() && vehicleId > 0 && vehicleId <= vehicles.size()) {  
 Renter renter = renters.get(renterId - 1);  
 Vehicle vehicle = vehicles.get(vehicleId - 1);  
  
 // Check if renter is eligible for the selected vehicle  
 if (vehicle.isEligible(renter.renterLicense)) {  
 System.*out*.println("The Renter " + renter.renterName + " is eligible for Vehicle " + vehicle.vehicleModel);  
 } else {  
 System.*out*.println("Sorry, you are not eligible for Vehicle " + vehicle.vehicleModel);  
 }  
 } else {  
 System.*out*.println("Invalid renter or vehicle ID.");  
 }  
 }  
  
}

package Q3;  
  
import java.util.Scanner;  
  
public class VehicleRenterApp {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Enter the Renter Name");  
 String renterName = scan.nextLine();  
 System.*out*.println("Enter the Renter License Type");  
 String renterLicenseType = scan.nextLine();  
 System.*out*.println("Enter the Renter Age");  
 int renterAge = scan.nextInt();  
 scan.nextLine();  
 RentalSystem rentalSystem = new RentalSystem();  
 Renter renter1 = new Renter(renterName,renterAge,renterLicenseType);  
 rentalSystem.registerRenter(renter1);  
 Renter renter2 = new Renter("Obaid",27,"Full");  
 rentalSystem.registerRenter(renter2);  
 Vehicle vehicle1=new Vehicle("Bike",1000,"Learner");  
 Vehicle vehicle2=new Vehicle("Car",25000,"Intermediate");  
 Vehicle vehicle3=new Vehicle("Truck",10000,"Full");  
 rentalSystem.registerVehicle(vehicle1);  
 rentalSystem.registerVehicle(vehicle2);  
 rentalSystem.registerVehicle(vehicle3);  
 rentalSystem.displayRenters();  
 System.*out*.println("Enter the Renter ID ");  
 int renterId = scan.nextInt();  
 scan.nextLine();  
 rentalSystem.displayVehicles();  
 System.*out*.println("Enter Vehicle ID");  
 int vehicleId = scan.nextInt();  
 scan.nextLine();  
 rentalSystem.rentVehicle(renterId,vehicleId);  
  
 }  
}

**Question 4:**

package Q4;  
  
public class Student {  
 String studentName;  
 int studentAge;  
 boolean feePaid;  
 boolean attendance=false;  
 public Student(String studentName, int studentAge) {  
 this.studentName = studentName;  
 this.studentAge = studentAge;  
 this.feePaid = false;  
 }  
 public String getStudentName() {  
 return studentName;  
 }  
 public void payTransportFee(){  
 this.feePaid = true;  
 }  
public void studentStatus(){  
 System.*out*.println("Name: "+this.studentName);  
 System.*out*.println("Age: "+this.studentAge);  
 if(feePaid){  
 System.*out*.println("Fee Paid: Yes");  
 }else{  
  
 System.*out*.println("Fee Paid: No");  
 }  
 System.*out*.println("Attendance: "+this.attendance);  
}  
}

package Q4;  
  
import java.util.ArrayList;  
  
public class BusRoute {  
 String routeName;  
 int totalStudents;  
 public void addBusRoute(String busRoute,int totalStudents) {  
 this.routeName = busRoute;  
 this.totalStudents=totalStudents;  
 }  
 void displayBusRoute() {  
 System.*out*.println("Route Name: "+this.routeName);  
 System.*out*.println("Present Students: "+this.totalStudents);  
 }  
 public String getBusRoute(){  
 return this.routeName;  
 }  
}

package Q4;  
  
import java.util.ArrayList;  
  
public class BusManager {  
 String managerName;  
 private final long busFee = 39500;  
 private static int *maxStudents*=50;  
 ArrayList<BusRoute> busRoutes;  
 ArrayList<Student> students;  
  
 public BusManager() {  
 this.managerName = "BusManager";  
 this.busRoutes = new ArrayList<>();  
 this.students = new ArrayList<>();  
 }  
public void stop(BusRoute busRoute) {  
 busRoutes.add(busRoute);  
}  
public void addStudent(Student student) {  
 students.add(student);  
}  
 public void registerBusRoute(String routeName, String studentName) {  
 boolean foundRoute = false;  
 boolean foundStudent = false;  
 for (BusRoute busRoute : busRoutes) {  
 if (busRoute.routeName.equals(routeName)) {  
 foundRoute = true;  
 busRoute.totalStudents++;  
  
 for (Student student : students) {  
 if (student.studentName.equals(studentName)) {  
 student.feePaid = true;  
 foundStudent = true;  
 System.*out*.println("Successfully registered for bus route " + busRoute.routeName);  
 break;  
 }  
 }  
 break;  
 }  
 }  
  
 if (!foundRoute) {  
 System.*out*.println("Bus Route " + routeName + " not found.");  
 } else if (!foundStudent) {  
 System.*out*.println("Student " + studentName + " not found.");  
 }  
 }  
  
  
 public void markAttendance(String studentName) {  
 for (Student student1 : students) {  
 int studentFound = 0;  
 if (student1.studentName.equals(student1.studentName)) {  
 studentFound = 1;  
 student1.attendance = true;  
 System.*out*.println("Successfully marked attendance for student " + student1.studentName);  
 break;  
 }  
 }  
 }  
  
 public void payFee(double fee,String routeName,String studentName) {  
 if (*maxStudents*<=50){  
 if (fee >= busFee) {  
 for (BusRoute busRoute : busRoutes) {  
 if (busRoute.routeName.equals(routeName)) {  
 for (Student student : students) {  
 if (student.studentName.equals(studentName)) {  
 student.feePaid = true;  
 }  
 }  
 busRoute.totalStudents++;  
 break;  
 }  
 }  
 double remaining=fee-this.busFee;  
 System.*out*.println("Fee is paid.Collect Remaining amount:Rs. "+remaining);  
 }else{  
 System.*out*.println("Insufficient fee");  
 }  
 }else{  
 System.*out*.println("Maximum number of students reached cannot ");  
 }  
 }  
}

package Q4;  
  
import java.util.Scanner;  
  
public class TransportationManagementSystem {  
 public static void main(String[] args) {  
 Scanner scan=new Scanner(System.*in*);  
 System.*out*.println("Enter the name of the student");  
 String studentName=scan.nextLine();  
 System.*out*.println("Enter the Age of the student");  
 int studentAge=scan.nextInt();  
 Student student2=new Student("Owais",18);  
 BusRoute busRoute1=new BusRoute();  
 busRoute1.addBusRoute("Clifton",34);  
 BusRoute busRoute2=new BusRoute();  
 busRoute2.addBusRoute("Nipa",48);  
 Student student1=new Student(studentName,studentAge);  
 BusManager manager1=new BusManager();  
 manager1.addStudent(student1);  
 manager1.addStudent(student2);  
 manager1.stop(busRoute1);  
 manager1.stop(busRoute2);  
 scan.nextLine();  
 System.*out*.println("Enter the Stop Name");  
 String stopName=scan.nextLine();  
 manager1.registerBusRoute(stopName,student1.studentName);  
 System.*out*.println("Enter the Amount you want to pay");  
 double amount=scan.nextDouble();  
 scan.nextLine();  
 System.*out*.println("Enter the Route name");  
 String routeName=scan.nextLine();  
 manager1.payFee(amount,routeName,student1.studentName);  
 System.*out*.println("Tap on the Student Name to mark Attendance");  
 String sName=scan.nextLine();  
 manager1.markAttendance(sName);  
 manager1.registerBusRoute(busRoute1.routeName,student2.studentName);  
 manager1.payFee(39500,"Clifton",student2.studentName);  
 System.*out*.println("Students Status");  
 student1.studentStatus();  
 System.*out*.println("Bus Route : "+busRoute2.getBusRoute());  
 student2.studentStatus();  
 System.*out*.println("Bus Route : "+busRoute1.getBusRoute());  
 }  
}