Assessment Cover Sheet Faculty of Engineering

Module Details			
Module Code	EC2492	**Module Title**	OOP
Program: SLIIT		**Course: BSc/ BEng/**	
Stream: EEE			

Assessment details			
Title	assignment	**Group assignment**	**NO**
		If yes, Group No.	
Lecturer/ Instructor		**Date of Performance**	
Due date	23rd of May 2021	**Date submitted**	23rd of May 2021

Student statement and signature

By this declaration, I/we confirm my/our understanding and acceptance that the work reported in this report is my/our own work. I/we also understand the consequences of engaging in plagiarism or copying others work without proper citation. Any material used in this work (whether from published sources, the internet or elsewhere) have been fully acknowledged and referenced and are without fabrication or falsification of data.

[Copying or plagiarism will result in a "0" mark for the continuous assessment and "F" for the module after an investigation on academic misconduct;

All academic misconduct is considered seriously and defined as dishonest and in direct opposition to the values of a learning community. Misconduct may result in penalties from failure to exclusion from the campus.
Further help and guidance on how to avoid academic misconduct can be obtained from your academic advisor/tutor]

By this declaration, I/we confirm my understanding and acceptance that-

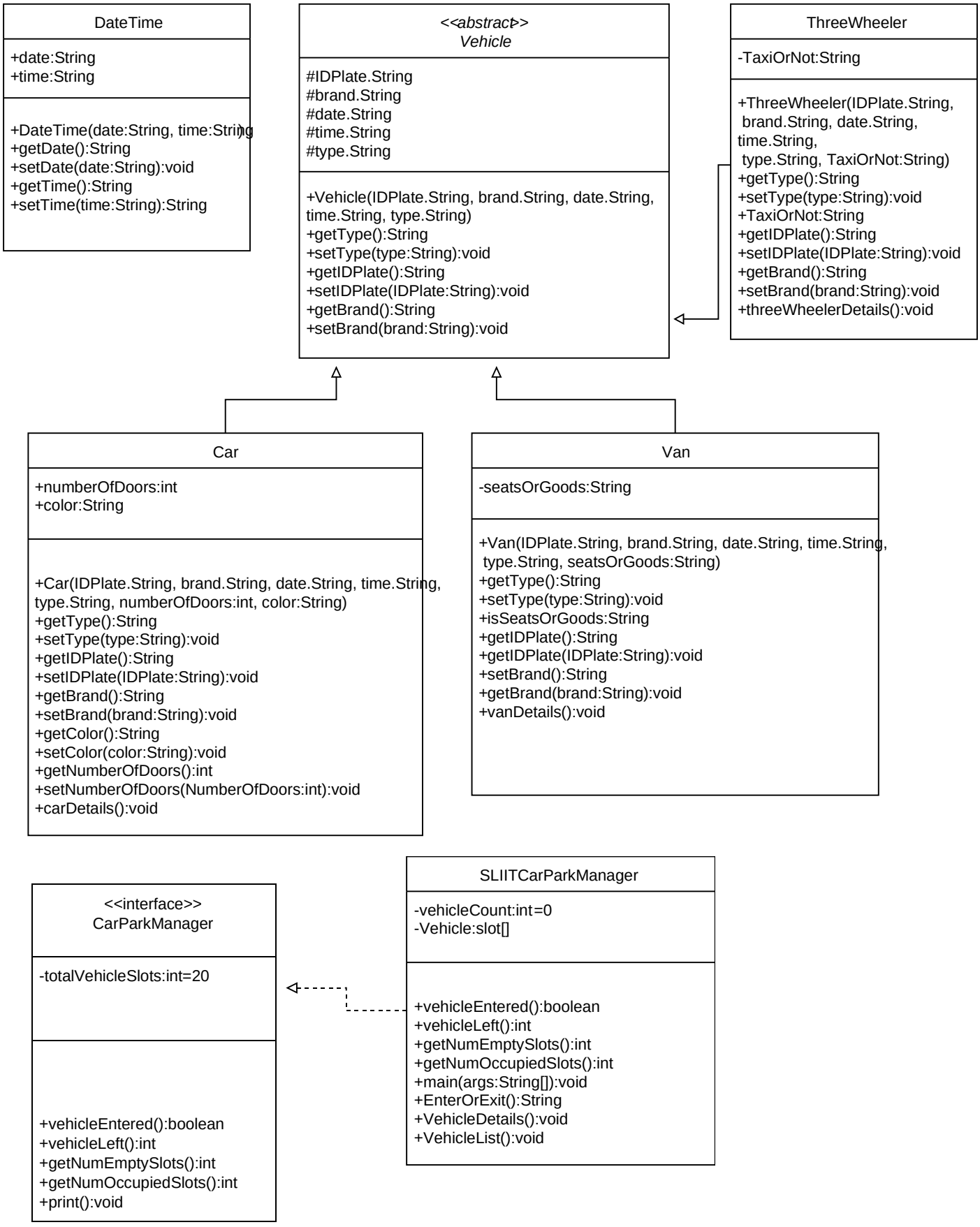
- **I/we have adhered to relevant ethical guidelines and procedures in the completion of the assignment.**
- **I/we have not allowed another student to have access to or copy from this work.**
- **This work has not been submitted previously.**

[The Institute may request an electronic copy of this work for submission to the Plagiarism detection facility (TURNITIN). You must make sure that an electronic copy of your work is available in these circumstances]

Details of the student/s submitting the assignment		Signature
ID Number	**Name (As per the institute records)**	
EN20412548	Dilshan R.G.A.	

Receiving Officer (seal, signature, date)	**Specific comments about the work (including overall comments and guidelines for improvement)**		
	Tutor:	**Signature:**	**Date:**
	Marks:	**[All marks are subject to external moderation and approval of board of examinations]**	

UML diagram


DateTime
+date:String
+time:String
+DateTime(date:String, time:String)
+getDate():String
+setDate(date:String):void
+getTime():String
+setTime(time:String):String
<<abstract>>
Vehicle
#IDPlate.String
#brand.String
#date.String
#time.String
#type.String
+Vehicle(IDPlate.String, brand.String, date.String, time.String, type.String)
+getType():String
+setType(type:String):void
+getIDPlate():String
+setIDPlate(IDPlate:String):void
+getBrand():String
+setBrand(brand:String):void
ThreeWheeler
-TaxiOrNot:String
+ThreeWheeler(IDPlate.String, brand.String, date.String, time.String, type.String, TaxiOrNot:String)
+getType():String
+setType(type:String):void
+TaxiOrNot:String
+getIDPlate():String
+setIDPlate(IDPlate:String):void
+getBrand():String
+setBrand(brand:String):void
+threeWheelerDetails():void
Car
+numberOfDoors:int
+color:String
+Car(IDPlate.String, brand.String, date.String, time.String, type.String, numberOfDoors:int, color:String)
+getType():String
+setType(type:String):void
+getIDPlate():String
+setIDPlate(IDPlate:String):void
+getBrand():String
+setBrand(brand:String):void
+getColor():String
+setColor(color:String):void
+getNumberOfDoors():int
+setNumberOfDoors(NumberOfDoors:int):void
+carDetails():void
Van
-seatsOrGoods:String
+Van(IDPlate.String, brand.String, date.String, time.String, type.String, seatsOrGoods:String)
+getType():String
+setType(type:String):void
+isSeatsOrGoods:String
+getIDPlate():String
+getIDPlate(IDPlate:String):void
+setBrand():String
+getBrand(brand:String):void
+vanDetails():void
<<interface>>
CarParkManager
-totalVehicleSlots:int=20
+vehicleEntered():boolean
+vehicleLeft():int
+getNumEmptySlots():int
+getNumOccupiedSlots():int
+print():void
SLIITCarParkManager
-vehicleCount:int=0
-Vehicle:slot[]
+vehicleEntered():boolean
+vehicleLeft():int
+getNumEmptySlots():int
+getNumOccupiedSlots():int
+main(args:String[]):void
+EnterOrExit():String
+VehicleDetails():void
+VehicleList():void

Code

Vehicle

```
abstract class Vehicle { //create a abstract class

    // create variables with protected accsess modifiers that can give
accsess to sub classes
    protected String IDPlate;
    protected String brand;
    protected String date;
    protected String time;
    protected String type;

    // create constructor
    public Vehicle(String type,String IDPlate,String brand,String
date,String time){
        this.IDPlate=IDPlate;
        this.brand=brand;
        this.date=date;
        this.time=time;
        this.type=type;
    }

    // gettype method for get thge type of the vehicle
    public String getType(){
        System.out.println(type);
        return type;
    }

    // settype of the vehicle for set the type of the vehicle (abstract
method)
    public abstract void setType(String type);

    public String getIDPlate(){
        System.out.println(IDPlate);
        return IDPlate;
    }
    // setIDplate of the vehicle for set the IDPlate of the vehicle
(abstract method)
    public abstract void setIDPlate(String IDPlate);

    // getbrand of the vehicle for get the brand of the vehicle (abstract
method)
    public abstract String getBrand();

    // setbrand of the vehicle for set the brand of the vehicle (abstract
method)
    public abstract void setBrand(String brand);

}
```

DateTime

```
public class DateTime { // create DateTime class

    // create variables with private accsess modifiers that can give
accsess to only own class
     private String date;
     private String time;

     // create constructor
    public DateTime(String date,String time){
        this.date=date;
        this.time=time;
    }

    // get method for get values
    public String getDate(){
        return date;
    }

    // set method for set values
    public void setDate(String date){
        this.date=date;
    }

    // get method for get values
    public String getTime(){
        return date;
    }

    // set method for set values
    public void setTime(String date){
        this.date=date;
    }

}
```

Car

```
public class Car extends Vehicle{   //car class inherit from Vehicle class

    // create variables with private accsess modifiers that can give
accsess to only own class
    private int NumberOfDoors;
    private String color;

    // create a constructor
    protected Car(String type,String IDPlate, String brand, String date,
String time,int NumberOfDoors,String color) {
        super(type,IDPlate, brand, date, time);
        this.IDPlate=IDPlate;
        this.brand=brand;
        this.date=date;
        this.time=time;
        this.NumberOfDoors=NumberOfDoors;
        this.color=color;
        this.type=type;


    }

    // get method for get values
    public String getType(){
        return type;
    }

    // get method for get values
    public void setType(String type){
        this.type=type;
    }

    // get method for get values
    @Override
    public String getIDPlate(){
        return IDPlate;
    }

    // set method for set values
    @Override
    public void setIDPlate(String IDPlate){
        this.IDPlate=IDPlate;
    }

    // get method for get values
    @Override
    public String getBrand(){
        return brand;
    }

    // set method for set values
    @Override
    public void setBrand(String brand){
        this.brand=brand;
    }

    // get method for get values
    public String getColor(){
        return color;
```

```
    }

    // set method for set values
    public void setColor(String color){
        this.color=color;
    }

    // get method for get values
    public int getNumberOfDoors(){
        return NumberOfDoors;
    }

    // set method for set values
    public void setNumberOfDoors(int NumberOfDoors){
        this.NumberOfDoors=NumberOfDoors;
    }

    // this nethod for print car deails
    public void carDetails(){
        System.out.println("\ndetails of your Vehicle,");
        System.out.println("ID Plate : "+IDPlate+"\nEntry Time :
"+time+"\nVehicle type is Car\n");
    }

}
```

Van

```
import java.util.Scanner;

public class Van extends Vehicle{   // van sub class extends with vehicle
super class

    // create variables with private accsess modifiers that can give
accsess to only own class
    private static String seatsOrGoods;

    // crate a constructor
    public Van(String type,String IDPlate, String brand, String date,
String time,String seatsOrGoods) {
        super(type,IDPlate, brand, date, time);
        this.IDPlate=IDPlate;
        this.brand=brand;
        this.date=date;
        this.time=time;
        this.seatsOrGoods=seatsOrGoods;
        this.type=type;

    }

    // get method for get values
    public String getType(){
        return type;
    }

    // set method for set values
    public void setType(String type){
        this.type=type;
    }

    // set method for set values
    public String isSeatsOrGoods(){
        Scanner scan=new Scanner(System.in);
        switch (seatsOrGoods){
            case "seats":
                System.out.println("Number Of Seats : ");
                String EnterSeats = scan.nextLine();
                break;

            case "goods":
                System.out.println("Goods Capacity(kg) : ");
                String EnterCapacity = scan.nextLine();
            break;
        }

        return seatsOrGoods;
    }

    // get method for get values
    @Override
    public String getIDPlate(){
        return IDPlate;
    }

    // set method for set values
    @Override
    public void setIDPlate(String IDPlate){
```

```
        this.IDPlate=IDPlate;
    }

    // get method for get values
    @Override
    public String getBrand(){
        return brand;
    }

    // set method for set values
    @Override
    public void setBrand(String brand){
        this.brand=brand;
    }

    // print van details
    public void vanDetails(){
        System.out.println("\ndetails of your Vehicle,");
        System.out.println("ID Plate : "+IDPlate+"\nEntry Time :
"+time+"\nVehicle type is Van\n");
    }
}
```

ThreeWheeler

```
import java.util.Scanner;

public class ThreeWheeler extends Vehicle{  // create 2wheel class extends
from Vehicle class

    // create variables with private accsess modifiers that can give
accsess to only own class
    private String TaxiOrNot;

    // craete a constructor
    public ThreeWheeler(String type,String IDPlate, String brand, String
date, String time,String TaxiOrNot) {
        super(type,IDPlate, brand, date, time);
        this.IDPlate=IDPlate;
        this.brand=brand;
        this.TaxiOrNot=TaxiOrNot;
        this.type=type;

    }

    // get method for get values
    public String getType(){
        return type;
    }

    // set method for set values
    public void setType(String type){
        this.type=type;
    }
    public String TaxiOrNot(){

        return TaxiOrNot;
    }

    // get method for get values
    @Override
    public String getIDPlate(){
        return IDPlate;
    }

    // set method for set values
    @Override
    public void setIDPlate(String IDPlate){
        this.IDPlate=IDPlate;
    }

    // get method for get values
    @Override
    public String getBrand(){
        return brand;
    }

    // set method for set values
    @Override
    public void setBrand(String brand){
        this.brand=brand;
    }

    // print details
```

```
    public void threeWheelerDetails(){
        System.out.println("\ndetails of your Vehicle,");
        System.out.println("ID Plate : "+IDPlate+"\nEntry Time :
"+time+"\nVehicle type is Three Wheeler\n");
    }
}
```

CarParkManager

```
public interface CarParkManager {   // create car park manager class as a
interface

    // created the methods and variables with nobody
    final int totalVehicleSlots = 20;
    boolean vehicleEntered();
    int vehicleLeft();
    int getNumEmptySlots();
    int getNumOccupiedSlots();
}
```

SLIITCarParkManager

```
import java.util.Scanner;

public class SLIITCarParkManager implements CarParkManager {    // crate
SLIITCarParkManager class, implement from CarParkManager class

    private static int vehicleCount=0;  // get a new variable to count
vehicles
    private static Vehicle []slot = new Vehicle[totalVehicleSlots]; //
crate an array to store the Vehicles

    private static boolean trueOrFalse; // crate a boolean to get the
return value from vehicle enter method

    // create the vehicle enter class for calculate entering vehicle
details
    @Override
    public boolean vehicleEntered() {
        if(vehicleCount<totalVehicleSlots){     // if for find space
available or not
            vehicleCount++; // incremant vehicle count if the space
available
                System.out.println("\n"+getNumEmptySlots()+" Slots
Available");//print the available slots
            trueOrFalse=true;//condition status
        }
        else {  // no space available
            System.out.println("Sorry, There are no parking slots
available\n");//a messege
            SLIITCarParkManager cm=new SLIITCarParkManager();// create a
object
            cm.EnterOrExit();// call the method

            trueOrFalse=false;//condition ststus
        }

        return trueOrFalse;// return status
    }

    // create a class for calculate regarding lefting vehicles
    @Override
    public int vehicleLeft() {
        Scanner scan=new Scanner(System.in);//create a scarnner object
        System.out.println("Your ID Plateeee :" );//asking a question
        String IDPlate = scan.nextLine();//get the answer and store the
data

    for(int i=0;i<=totalVehicleSlots;i++){ // looking for the leaving
vehicle array slots
        if(slot[i]==null) { //fix the error if find a empty array
           // System.out.println("slot "+i+" is empty");

        }else {
            String searchIDPlate = slot[i].getIDPlate();// leaving
vehicle`s number plate

            if (IDPlate.equals(searchIDPlate)) {// compare number plates
                vehicleCount--; // decremant vehicles
```

```
                System.out.println("\n"+getNumOccupiedSlots() + " free
vehicle Slots available\n");
                slot[i] = null; // make thew slot empty
                vehicleList(); // go to the vehicle details
                break;  // break the loop
            }

        }

    }

        return 0 ;
    }

    //calculate empty slots
    @Override
    public int getNumEmptySlots() {
        return totalVehicleSlots-vehicleCount +1;
    }

    //calculate occupied slots
    @Override
    public int getNumOccupiedSlots() {
        return totalVehicleSlots-vehicleCount;
    }

    // main method
    public static void main(String[] args) {

        // create a infinite loop
        while (true) {

            Scanner scan = new Scanner(System.in);  //create scanner object
            SLIITCarParkManager cm = new SLIITCarParkManager();//create
object

            // switch statement for find the vehicle enter or exit
            switch (cm.EnterOrExit()){

                case "enter":   // if vehicle enter do the followings
                    cm.vehicleEntered(); // go to this class
                    cm.VehicleDetails(); // go to this class
                    break;  // break the switch statement

                case "exit": // if vehicle exit do the followings
                    cm.vehicleLeft();// go to this class
                    break;// break the switch statement
            }
        }
    }

    // create a method
    public String EnterOrExit(){
        Scanner scan=new Scanner(System.in);    //create object
        System.out.println("Enter or Exit : "); //ask a question
        String EnterExit = scan.nextLine(); // store the answer

        return EnterExit;   //return the answer
    }

    // vehicle details class
    public void VehicleDetails(){
```

```
        Scanner scan=new Scanner(System.in);     //create object
        System.out.println("Vehicle Type (car,van,three wheel): "); //ask a
question
        String vehicle_type = scan.nextLine();  // create a object
        // if condition for get the correct input

if(vehicle_type.equals("car")||vehicle_type.equals("van")||vehicle_type.equ
als("three wheel")) {
            switch (vehicle_type) { //switch statement
                case "car":

                    /*
                    System.out.println("xxxxxxx"); ----------> asking a
question
                    String carIDPlate = scan.nextLine(); ----> store it
                    */

                    System.out.println("ID Plate : ");
                    String carIDPlate = scan.nextLine();

                    System.out.println("Brand : ");
                    String carbrand = scan.nextLine();

                    System.out.println("Date : ");
                    String cardate = scan.nextLine();

                    System.out.println("Time : ");
                    String cartime = scan.nextLine();

                    System.out.println("Color : ");
                    String color = scan.nextLine();

                    System.out.println("Number of doors : ");
                    int NumberOfDoors = scan.nextInt();

                    // create a object for car constructor
                    Car c = new Car(vehicle_type,carIDPlate, carbrand,
cardate, cartime, NumberOfDoors, color);
                    c.carDetails(); // call to the car details

                    // for loop for store the details in the array
                    for (int i = 0; i <= 20; i++) {
                        if (slot[i] == null) {  //find a empty slot
                            slot[i] = c;    // assign values
                            break;  // break the loop
                        }

                    }
                    break;

                case "van":
                    System.out.println("ID Plate : ");
                    String vanIDPlate = scan.nextLine();

                    System.out.println("Brand : ");
                    String vanbrand = scan.nextLine();

                    System.out.println("Date : ");
                    String vandate = scan.nextLine();

                    System.out.println("Time : ");
                    String vantime = scan.nextLine();
```

```
                    System.out.println("Seats or Goods : ");
                    String seatsOrGoods = scan.nextLine();
                    //create a object
                    Van v = new Van(vehicle_type,vanIDPlate, vanbrand,
vandate, vantime, seatsOrGoods);
                    v.isSeatsOrGoods(); // fint the van is goods or seats
                    v.vanDetails(); // call to print details

                    // for loop for store the details in the array
                    for (int i = 0; i <= 20; i++) {
                        if (slot[i] == null) {  // find a empty slot
                            slot[i] = v;    // assign values
                            break;//break loop
                        }

                    }
                    break;// break switch

                case "three wheel":
                    System.out.println("ID Plate : ");
                    String twlIDPlate = scan.nextLine();

                    System.out.println("Brand : ");
                    String twlbrand = scan.nextLine();

                    System.out.println("Date : ");
                    String twldate = scan.nextLine();

                    System.out.println("Time : ");
                    String twltime = scan.nextLine();

                    System.out.println("Taxi or Not for hire :");
                    String taxiOrNotForHire = scan.nextLine();

                    // create a object
                    ThreeWheeler t = new
ThreeWheeler(vehicle_type,twlIDPlate, twlbrand, twldate, twltime,
taxiOrNotForHire);
                    t.threeWheelerDetails();

                    // for loop for store the details in the array
                    for (int i = 0; i <= 20; i++) {
                        if (slot[i] == null) {  // find a empty slot
                            slot[i] = t;    //aassign values
                            break;  // break for loop
                        }

                    }
                    break;  // break switch statement
            }
        }else { // if the enterd vehicle type is wrong
            System.out.println("\nPlease enter correct key word
again\n");// print a messege
            vehicleCount--;//decrement vehicle count for avoid a error
            SLIITCarParkManager cm=new SLIITCarParkManager(); // crewate
object
            cm.EnterOrExit();   // call to this method

        }
    }
```

```
    // this class for print vehicle list
    public void vehicleList(){
        System.out.println("currently parked vehicle list,");//print a
message
        for(int i=0;i<totalVehicleSlots;i++){    // for lopo for find
vehicle slots
            if(slot[i]==null){  // avoid the eroor come with null classes
            //System.out.println("Vehicle slot "+(i+1)+" is empty");
            }
            else {
                // print the detail list
                System.out.println("ID Plate:"+slot[i].getIDPlate()+",
Entry Time:"+slot[i].time+", Vehicle type is:"+slot[i].getType());
                }
        }
        System.out.println();// just break a line
    }
}
```

Demo video link

https://drive.google.com/drive/folders/1-TYI7MsFcGSniqdsTxHji4CwMFzoRrB9?usp=sharing