SLIIT
Discover Your Future

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Topic : Automated Parking System

Group no: MLB_WD_01.01_09

Campus: Malabe

Submission Date: 2023/10/31

We declare that this is our own work, and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT22111692	ABEWARDHANA J.H. K	077 5082841
IT22272768	BASNAYAKA W.B.M.D. K	076 7672382
IT22127778	CHAVINDEE M.A. P	071 2174880
IT22138668	PERERA T.M. S	077 9423127
IT22116260	THARINDI W.A. K	076 1758678

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

1.Description of the requirements

- Guests can get register to the system by providing required details for the registration.
- Registered User login to the system using by providing required user login credential.
- Registered User can Contact Arrow Automated parking system quest in the contact number of the website.
- Registered User can Edit their own Account details.
- System stores vehicle details and displays to the customer.
- System manager generate summary report by ensuring quality of overall system
- Users can inform their request and problems by using support page.
- Users can visit web site and choose their payment package and do payment online or bank deposits.
- After payment, user can get their parking tickets.
- Through the web site user can reserve their parking location and time.
- User can review guidelines & get an idea about privacy & polices.

2.Classes Identified

- Guest
- Customer
- Vehicle
- Parking
- Report
- Administrator
- Payment
- Guideline
- Support
- Complain
- Parking ticket

3. CRC card

Guest	
Responsibilities	**Collaboration**
Give correct required details	Registration

Customer	
Responsibilities	**Collaboration**
Providing correct credentials	Registration
Edit Account details	Registration
Borrow ticket	Book ticket
Pay fine	

Book ticket	
Responsibilities	**Collaboration**
Confirm ticket	Payment

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Payment	
Responsibilities	**Collaboration**
Store payment details	
Validate payment	

Registration	
Responsibilities	**Collaboration**
Store details of customers	
Update Details	
Validate	

Report	
Responsibilities	**Collaboration**
List of ticket	Book ticket
List of previous booking	Book ticket, Customer

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Guideline	
Responsibilities	**Collaboration**
Include system rules and regulations	Customer

Vehicle	
Responsibilities	**Collaboration**
Display vehicle details	
Confirm vehicle authority	Customer
Store vehicle details	
Get parking details	Parking

Parking	
Responsibilities	**Collaboration**
Display parking details	
Store parking details	
Check vehicle number	Vehicle

Complain	
Responsibilities	**Collaboration**
Display complains details	
Store complain details	

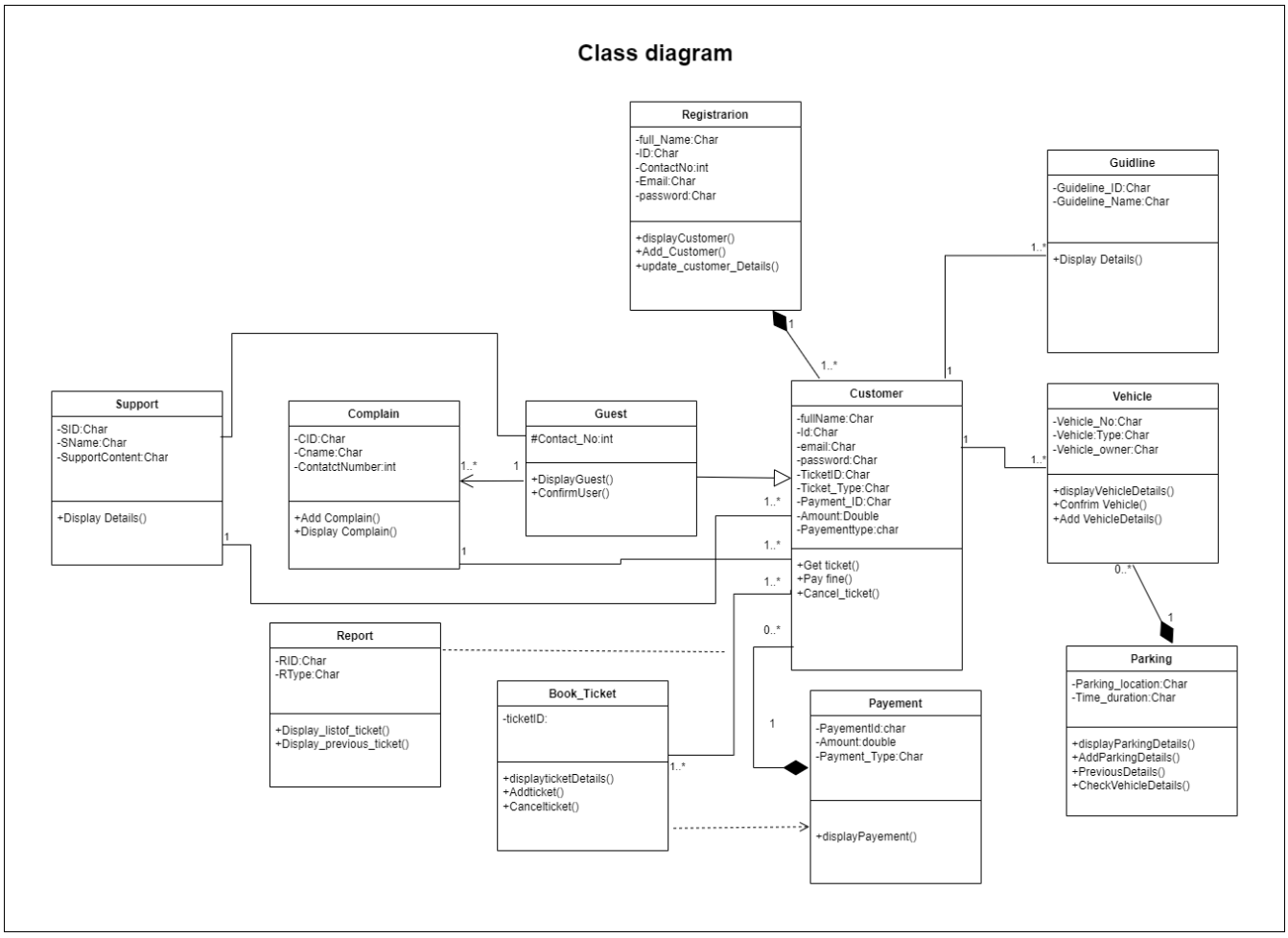
Support	
Responsibilities	**Collaboration**
Display Support details	
Store support details	

Book ticket	
Responsibilities	**Collaboration**
Display ticket details	
Cancel ticket	

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**



BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Main.CPP

```
// automated parking system.cpp : This file contains the 'main' function. Program execution
begins and ends there.
#include "Book_Ticket.h"
#include "Complain.h"
#include "customer.h"
#include "Registration.h"
#include "Guideline.h"
#include "payment.h"
#include "Parking.h"
#include "Report.h"
#include "Support.h"
#include "Vehicle.h"
#include<iostream>
#include<cstring>

using namespace std;

int main()
{
    //customer & guest
    Customer c1, c2, c3;
    Guest g1;

    c1.assingCustomer("Hashen", "ID001", "hashenkavishka60@gmail.com", "Hashen123");
    c2.assingCustomer("Maheli", "ID002", "Maheli60@gmail.com", "Hashen123");
    c3.assingCustomer("pasindya", "ID003", "pasindya60@gmail.com", "Hashen123");
    c1.displaycustomer();
    c2.displaycustomer();
    c3.displaycustomer();



    //create varibale
    int Gcontect;

    //user input
    cout << "Enter number: ";
    cin >> Gcontect;

    //dispaly customer details
    g1.assingContactNo(Gcontect);
    g1.displayguest();

    //Whole and part composition
    Registraction* r1;
    r1 = new Registraction();
    r1->displaycustomer();
    delete r1;
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

```
//guideline & customer


 Guideline* CG1 = new Guideline("Praivacy & Policy", "customer must need to agree this
privacy & policy");

 Customer* gc1 = new Customer("KAVISHKA", "ID001", "hashenkavishka60@gmail.com",
"Hashen123", "TID001", "Nomal", "PID001", 2000, "Nomal", CG1);


    Customer* vc1 = new Customer("WPS 1200", "Axio car ", "J.H.K.Perera");
    Customer* vc2 = new Customer("Www 1000", "Axio car ", "J.H.K.Pasindaya");

    //parking
    Parking* park = new Parking("1st floor 2nd rack", "8.00 a.m-5.00p.m");
    cout << "Parking Details:" << endl;
    park->displayParkingDetails();
    cout << endl;

    //payment
    Payment* p1 = new Payment(001, 6000.00, "Bankdeposite");
    cout << "Payment details : " << endl;
    p1->displayPayment();
    cout << endl;

    //complain
    Complain* co = new Complain("C001", "Poor Service");
    Guest* d = new Guest("C002", "Billing Issue");

    //vehicle
    Customer* VC;
    Vehicle* VC1;

    //suport


    return 0;
}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Customer.h

```
#pragma once
#include "Complain.h"
#include "Support.h"
#include "Guideline.h"
#include "vehicle.h"
#include "Vehicle.h"
#include "Book_Ticket.h"
class Customer
{
protected:
	char fullName[50];
	char Id[20];
	char email[50];
	char password[100];
	char TicketID[50];
	char Ticket_Type[50];
	char Payment_ID[50];
	double Amount;
	char Paymenttype[50];
	Guideline* cus;
	Support* sc;
	Book_Ticket* BC;
	Vehicle* V[2];
public:
	Customer();
	Customer(const char FULLname[], const char iD[], const char eMail[], const char
pAssword[], const char tIcketid[], const char tIckete_type[], const char pAyment_ID[],
double aMount, const char PAymenttype[], Guideline* cus1);
	void get_ticket(const char TICKETID[], const char  TICKET_TYPE[]);
	void payfine(const char paymentid[], double amount, const char paymenttype[]);
	void cancle_ticket(const char  TIcketId[], const char  TIcket_type[]);
	void assingCustomer(const char  FULLNAME[], const char  ID[], const char EMAIL[],
const char  PASSWORD[]);
	void displaycustomer();
	~Customer();
};

class Guest : public Customer {
protected:
	int contact_no;
	Complain* cg;
	Support* cs;
public:
	Guest();
	void assingContactNo(int CONTACT_NO, Complain* cg1);
	void addsupport(int CONTACT_No, Support* cs1);
	void displayguest();
};
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Customer.Cpp

```
#include "customer.h"
#include<iostream>
#include<cstring>
#include "Vehicle.h"
#include "Guideline.h"
#include "Support.h"
using namespace std;

//peremeter
Customer::Customer()
{
	strcpy_s(fullName, "");
	strcpy_s(Id, "");
	strcpy_s(email, "");
	strcpy_s(password, "");
	strcpy_s(TicketID, "");
	strcpy_s(Ticket_Type, "");
	strcpy_s(Payment_ID, "");
	Amount = 0;
	strcpy_s(Paymenttype, "");

}
Customer::Customer(const char FULLname[], const char iD[], const char eMail[], const char
pAssword[], const char tIcketid[], const char tIckete_type[], const char pAyment_ID[],
double aMount, const char PAymenttype[], Guideline* cus1)
{
	strcpy_s(fullName, FULLname);
	strcpy_s(Id, iD);
	strcpy_s(email, eMail);
	strcpy_s(password, pAssword);
	strcpy_s(TicketID, tIcketid);
	strcpy_s(Ticket_Type, tIckete_type);
	strcpy_s(Payment_ID, pAyment_ID);
	Amount = aMount;
	strcpy_s(Paymenttype, PAymenttype);
	cus = cus1;
	cus->addguideline(this);

}
void Customer::get_ticket(const char TICKETID[], const char  TICKET_TYPE[])
{
	strcpy_s(TicketID, TICKETID);
	strcpy_s(Ticket_Type, TICKET_TYPE);
}
void Customer::payfine(const char paymentid[], double amount, const char  paymenttype[])
{
	strcpy_s(Payment_ID, paymentid);
	Amount = amount;
	strcpy_s(Paymenttype, paymenttype);
```

```
}



void Customer::cancle_ticket(const char  TIcketId[], const char TIcket_type[])
{
	strcpy_s(TicketID, TIcketId);
	strcpy_s(Ticket_Type, TIcket_type);
}

void Customer::assingCustomer(const char FULLNAME[], const char  ID[], const char  EMAIL[],
const char  PASSWORD[])
{
	strcpy_s(fullName, FULLNAME);
	strcpy_s(Id, ID);
	strcpy_s(email, EMAIL);
	strcpy_s(password, PASSWORD);

}

void Customer::displaycustomer()
{
	cout << "Customer details" << endl;

	cout << "Full name: " << fullName << endl;
	cout << "ID: " << Id << endl;
	cout << "Email: " << email << endl;
	cout << "password: " << password << endl;
	cout << "---------------------------------" << endl;
}


//customer destructor
Customer::~Customer()
{
	cout << "Delete customer :" << fullName << Id << email << password << endl;
}



void Guest::assingContactNo(int CONTACT_NO, Complain* cg1)
{
	contact_no = CONTACT_NO;
	cg = cg1;
}

void Guest::addsupport(int CONTACT_No, Support* cs1)
{
	contact_no = CONTACT_No;
	cs = cs1;
}
void Guest::displayguest()
{
```

```
	cout << "Gust Details" << endl;
	cout << "Contact number" << contact_no << endl;
}
Guest::Guest()
{
	contact_no  = 0;
}
```

Registration.h

```
#pragma once
#include "customer.h"
class Registraction //whole
{
private:
	Customer* cust[2];

public:
	Registraction();//Default castrater
	Registraction(Customer* cust1[]);//Overloard  castrater
	void displaycustomer();
	~Registraction();//Destructor
};
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Registration.cpp

```
#include "customer.h"
#include "Registration.h"
#include<iostream>
#include<cstring>
using namespace std;
Registraction::Registraction()
{
	cust[0] = new Customer("kavisha", "ID004", "hashenkavishka60@gmail.com",
"kavisha123", "t001", "nomal", "p001", 200, "online");
	cust[1] = new Customer("Lashan", "ID005", "hashenkavishka60@gmail.com", "lashan123",
"t002", "nomal", "p002", 2000, "online");
}

Registraction::Registraction(Customer* cust1[])
{
	for (int i = 0; i < 2; i++) {
		cust[i] = new Customer(*cust1[i]);  // Assuming Customer class has a copy
constructor
	}
}


void Registraction::displaycustomer()
{
	cout << "customer details:" << endl;
	for (int i = 0; i < 2; i++) {
		cust[i]->displaycustomer();
	}

}

Registraction::~Registraction()
{
	cout << "Customer details" << endl;
	for (int i = 0; i < 2; i++) {
		delete cust[i];
	}
}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Book Ticket.h

```
#pragma once
#include "customer.h"
#include<iostream>
#include<cstring>

class Book_Ticket
{
private:
	char ticketID[20];
	Customer* BC[2];
public:
	Book_Ticket(const char TicketID[]);
	void Addticket(const char TIcketID[], Customer* BC1);
	void displayticketDetails(const char TIcketID[]);
	~Book_Ticket();
};
```

Book Ticket.cpp

```
#include "Book_Ticket.h"
#include<cstring>
#include<iostream>
#include<cstring>
using namespace std;

//ticket ccp
#include<cstring>
using namespace std;

Book_Ticket::Book_Ticket(const char TicketID[])
{
	strcpy_s(ticketID, TicketID);
}
void Book_Ticket::displayticketDetails(const char TIcketID[])
{
	cout << "Ticket ID:" << ticketID << endl;

}
void Book_Ticket::Addticket(const char TIcketID[], Customer* BC1)
{

}
Book_Ticket::~Book_Ticket()
{
	cout << "Deleted" << ticketID  << endl;
}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Complain.h

```
#pragma once
#include "Complain.h"
#include "Support.h"
#include "Guideline.h"
#include "vehicle.h"
#include "Vehicle.h"
#include "Book_Ticket.h"
class Customer
{
protected:
	char fullName[50];
	char Id[20];
	char email[50];
	char password[100];
	char TicketID[50];
	char Ticket_Type[50];
	char Payment_ID[50];
	double Amount;
	char Paymenttype[50];
	Guideline* cus;
	Support* sc;
	Book_Ticket* BC;
	Vehicle* V[2];
public:
	Customer();
	Customer(const char FULLname[], const char iD[], const char eMail[], const char
pAssword[], const char tIcketid[], const char tIckete_type[], const char pAyment_ID[],
double aMount, const char PAymenttype[], Guideline* cus1);
	void get_ticket(const char TICKETID[], const char  TICKET_TYPE[]);
	void payfine(const char paymentid[], double amount, const char paymenttype[]);
	void cancle_ticket(const char  TIcketId[], const char  TIcket_type[]);
	void assingCustomer(const char  FULLNAME[], const char  ID[], const char EMAIL[],
const char  PASSWORD[]);
	void displaycustomer();
	~Customer();
};

class Guest : public Customer {
protected:
	int contact_no;
	Complain* cg;
	Support* cs;
public:
	Guest();
	void assingContactNo(int CONTACT_NO, Complain* cg1);
	void addsupport(int CONTACT_No, Support* cs1);
	void displayguest();
};
```

Complain.cpp

```
#include "Complain.h"
#include <cstring>
#include <iostream>

using namespace std;

int Complain::Contactnumber()
{
	return Contactnumber();
}

void Complain::AddComplain(const char cid[], const char cname[])
{
	strcpy_s(ComplainID, cid);
	strcpy_s(Complainame, cname);
}

void Complain::DisplayComplain()
{
	cout << ComplainID << Complainame << endl;
}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Guideline.h

```
#pragma once
#include "customer.h"
#define SIZE 2
class Guideline
{
private:
	char GuidelineName[30];
	char GuidelineContent[100];
	Customer* c1[SIZE];

public:
	Guideline() {};
	Guideline(const char name[], const char con[]);
	void addguideline(Customer* C);
	void displayGuideline();
	~Guideline();
};
```

Guideline.cpp

```
#include "Guideline.h"
#include "customer.h"
#include<cstring>
#include<iostream>
using namespace std;

//default constructer for paymnet
//Guideline::Guideline()
//{
    //strcpy(GuidelineName, "");
//    strcpy(GuidelineContent, "");
//}


//constructer with parameters
Guideline::Guideline(const char name[], const char con[])
{
    strcpy(GuidelineName, name);
    strcpy(GuidelineContent, con);

}
void Guideline::displayGuideline()
{
    cout << GuidelineName << GuidelineContent << endl;
}
```

```
void Guideline::addguideline(Customer* C)
{
    for (int i = 0; i < SIZE; i++) {
        if (c1[i] == nullptr) {
            c1[i] = C;
            break;
        }
    }
}

Guideline::~Guideline()
{
    for (int i = 0; i < SIZE; i++) {
        if (c1[i] != nullptr) {
            delete c1[i];
        }
    }
}
```

Parking.h

```
#pragma once
#include"Vehicle.h"
#define SIZE 2
class Parking
{
private:
	char Parking_location[40];
	char parking_duration[30];
	Vehicle* vehi[SIZE];
public:
	Parking();
	Parking(char const P_location[], char const P_duration[]);
	void displayParkingDetails();
	void AddParkingDetails();
	void previousDetails();
	void checkVehicleDetails();

	void showvehi();
	void checkvehi(char Vehicle_No[], char Vehicle_Type[], char Vehicle_owner[]);

	~Parking();
};
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Parking.cpp

```
#include "Parking.h"
#include"Vehicle.h"
#include<cstring>
#include<iostream>

using namespace std;

//default constructor for parking
Parking::Parking()
{
	strcpy_s(Parking_location, "");
	strcpy_s(parking_duration, "");
}
//constructor with parameters
Parking::Parking(char const P_location[], char const P_duration[]) {
	strcpy_s(Parking_location, P_location);
	strcpy_s(parking_duration, P_duration);

}
void Parking::displayParkingDetails() {
	Parking::displayParkingDetails();
	cout << Parking_location << endl << parking_duration << endl;

}
void Parking::AddParkingDetails() {

}
void Parking::previousDetails() {

}
void Parking::checkVehicleDetails() {

}
//vehicle part in parking (composition)
void Parking::checkvehi(char Vehicle_No[], char Vehicle_Type[], char Vehicle_owner[]) {
	vehi[0] = new Vehicle(Vehicle_No, Vehicle_Type, Vehicle_owner);
}
void Parking::showvehi() {

	vehi[0]->displayVehicledetails();
}


Parking::~Parking() {
}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Payment.h

```
#pragma once
#include"Customer.h"
#define SIZE 2
class Payment
{
private:
	char paymentID[10];
	double amount;
	char paymentType[30];
	Customer* C1[SIZE];

public:
	Payment();
	Payment(char id[], double amt, char payT[]);
	void displayPayment();
	~Payment();
};
```

Payment.cpp

```
/ #include "Bookticket"
#include<cstring>
#include "payment.h"
#include<iostream>
using namespace std;

//default constructer for paymnet
Payment::Payment()
{
	strcpy(paymentID, "");
	amount, "";
	strcpy(paymentType, "");
}

//constructer with parameters
Payment::Payment(char id[], double amt, char payT[])
{
	strcpy(paymentID, id);
	amount = amt;
	strcpy(paymentType, payT);
}
void Payment::displayPayment()
{
	Payment::displayPayment();
}
Payment::~Payment()
{

}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Report.h

```
#include <cstring>
class Report {
private:
	char reportId;
	char reportType;
	int reportDate;

public:
	void setlistOfTickets(const char RID[], const char RType[], int Rdate);
	void displaypreviousTicket();
};
```

Report.cpp

```
//Report-main.cpp
#include<iostream>
#include "Report.h"
#include <cstring>
using namespace std;

void Report::setlistOfTickets(const char RID[], const char RType[], int Rdate)
{
	strcpy_s(reportId, RID);
	strcpy_s(reportType, RType);
	reportDate = Rdate;
}

void Report::displaypreviousTicket()
{
	cout << "Report Id: " << reportId << endl;
	cout << "Report Type: " << reportType << endl;
	cout << "Report Date: " << reportDate << endl  << endl;
}
```

Support.h

```
#pragma once
```

```
#include "customer.h"
class Support
{
private:

	char supportID[20];
	char supportName[20];
	char supportContact[20];
	Customer* sc[2];
public:
	Support();
	Support(const char sid[], const char sname[], const char sco[]);
	void addsupport(Customer* sc1);
	void displayDetails();

};
```

Support.cpp

```
#include "Support.h"
#include <iostream>
#include <cstring>

using namespace std;

Support::Support()
{
    strcpy_s(supportID, "");
    strcpy_s(supportName, "");
    strcpy_s(supportContact, "");
}

Support::Support(const char sid[], const char sname[], const char sco[])
{
    strcpy_s(supportID, sid);
    strcpy_s(supportName, sname);
    strcpy_s(supportContact, sco);
}
void Support::addsupport(Customer* sc1)
{
    sc = sc1;
}

void Support::displayDetails()
{
    cout << supportID << supportName << supportContact << endl;
}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2 Year 1, Semester 2 **2023-July-Dec**

Vehicle.h

```
#include "customer.h"
#pragma once


class Vehicle {
private:
	char Vehicle_No[20];
	char Vehicle_Type[30];
	char Vehicle_owner[30];
	Customer* cus;
public:
	Vehicle();
	Vehicle(char const V_No[], char const  V_Type[], char const V_owner[]);
	void addvehicle(const char VehicleNo[], const char Vehicletype[], const char
vehicleowner[], Customer* pcus);
	void displayVehicledetails();
	~Vehicle();
};
```

Vehicle.cpp

```
//vehicle cpp
#include "Vehicle.h"
#include<cstring>
#include<iostream>
using namespace std;

Vehicle::Vehicle() //default constructor implement
{
	strcpy_s(Vehicle_No, "");
	strcpy_s(Vehicle_Type, "");
	strcpy_s(Vehicle_owner, "");
}

Vehicle::Vehicle(char const V_No[], char const V_Type[], char const V_owner[])
//overloaded constructor implementation
{
	strcpy_s(Vehicle_No, V_No);
	strcpy_s(Vehicle_Type, V_Type);
	strcpy_s(Vehicle_owner, V_owner);
}
void Vehicle::displayVehicledetails() {
	cout << Vehicle_No << endl << Vehicle_Type << endl << Vehicle_owner << endl;
```

```
}
void confrimVhicle() {
}
void Vehicle::addvehicle(const char VehicleNo[], const char Vehicletype[], const char
vehicleowner[], Customer* pcus)
{
	strcpy_s(Vehicle_No, VehicleNo);
	strcpy_s(Vehicle_Type, Vehicletype);
	strcpy_s(Vehicle_owner, vehicleowner);
	cus = pcus;
}

Vehicle::~Vehicle()//destructors implementation
{

}
```