Topic : Online Fitness Trainer

Group no : MLB_09.01_04

Campus : Malabe

Submission Date :

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**
IT21168222	Saranga G K G	0718535570
IT21170348	Dinujaya K G M	0774376203
IT21168604	Weihena K M	0715943250
IT21171710	Oshadari K M R P	0778250538
IT21171406	Ekanayake E P M T S	0768295683

Exercise 1:

Requirements

- An unregistered user/ new customer to become a member in the online fitness training system needs to register providing details such as name, e-mail address.
- A registered customer should log in to the website through his/her user account by using credentials.
- The system should be able to validate user
- A registered customer should be able to update their account details
- A registered customer can check their membership status
- The registered user selects and subscribes a membership package by payment through an acceptable method (PayPal, VISA, Master)
- The registered user confirms the payment is validation date and expiration date for the package.
- The system should be able to validate memberships
- The registered user can join the sessions conducted by trainers.
- The customer can add items to the cart
- The customer can buy fitness items through the online store.
- The admin can update package details.
- Admin generates the financial report, trainer report and customer report.
- A trainer can create sessions and update session details on the package page.
- The system should be able to get customer's feedback
- System Admin should be able to reply to feedback

Identified classes

- Customer (Registered)
- User Account
- MembershipPackage
- Subscription
- Payment
- Cart
- Item
- Trainer
- Session
- Report
- Feedback

CRC Cards

IT21168222

Name : Customer	
Responsibility	**Collaborator**
Store customer details	
Log in to system	User account
Check membership status	Subscription
Subscribe a package	Payment, membershipPackage, Subscription
Buy items	Payment, item, cart
Give feedback	Feedback

Name : User Account	
Responsibility	**Collaborator**
Give permission to access the services	
Store credentials	

Name : Subscription	
Responsibility	**Collaborator**
Store subscription details	
Renew Subscription	

IT21170348

Package	
Responsibilities	**Collaborations**
Display package details	
Add session time	trainer

Payment	
Responsibilities	**collaboration**
Validate	
Store payment details	

IT21171710

Item	
Responsibilities	**Collaboration**
Store item details	

Trainer	
Responsibilities	**Collaboration**
Create session	Session
Update session details	Package
Conduct session	

IT21171406

Session	
Responsibilities	**Collaboration**
Display session details	

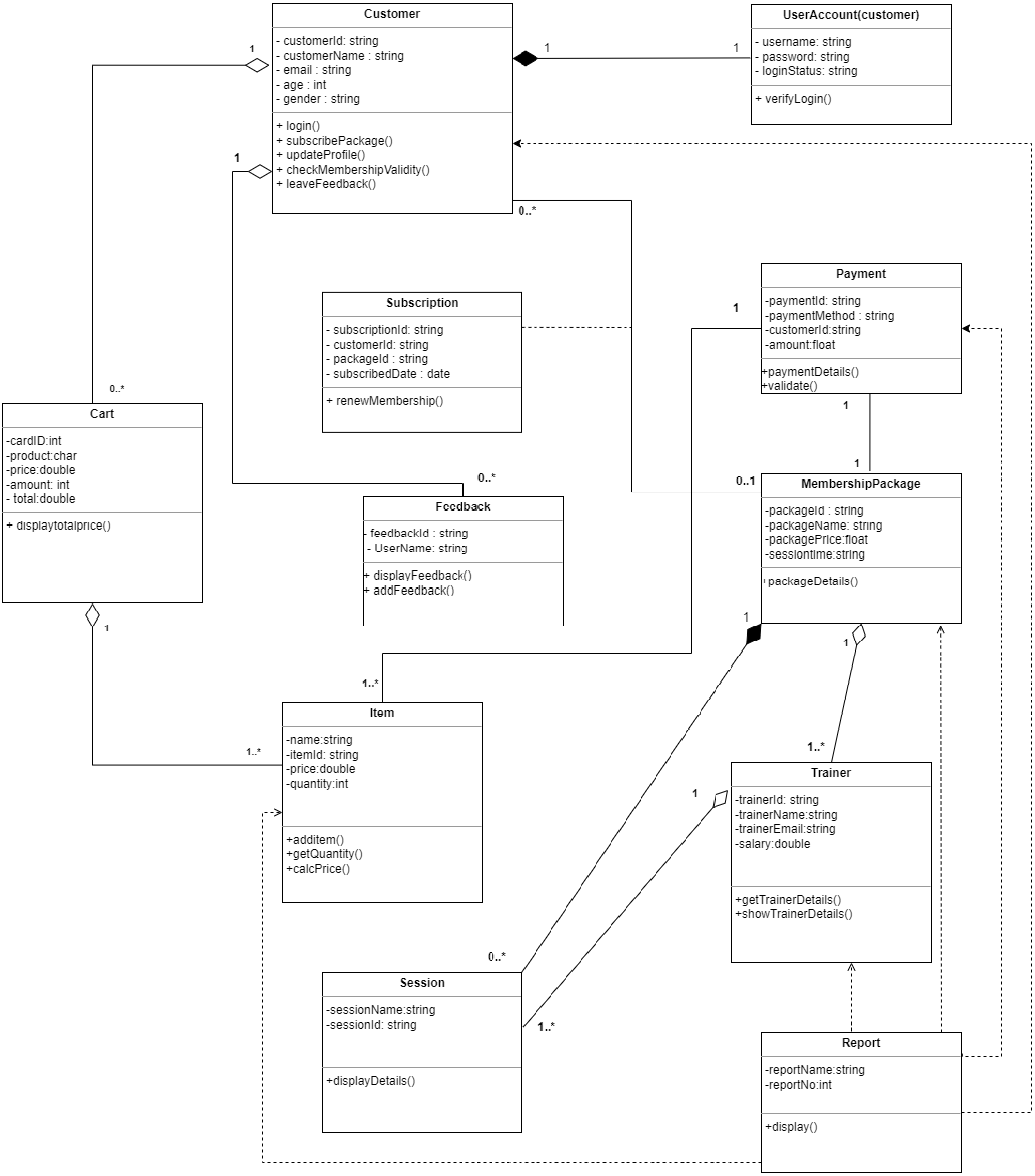
Report	
Responsibilities	**Collaboration**
Store payment details	Payment
List of customer details	Customer
List of trainer details	Trainer

IT21168604

Feedback	
Responsibility	**Collaboration**
Display feedbacks	
Store feedbacks	

Cart	
Responsibility	**Collaboration**
Store total price	
Display total price	item

Class Diagram



--- IT21168222 ---

```
//These codes are written with a focus on one object
#include <iostream>
#include <string>
using namespace std;

class UserAccount {
private:
	string username;
	string password;
	string loginStatus;
public:
	UserAccount() {};
	UserAccount(string uName) {
		username = uName;
	}
	void Display() {
		cout << "User Name " << username << endl;
	}
	~UserAccount() {
		cout << "Deleting user " << username << endl;
	}
};

class MembershipPackage {
private:
        string membershipId;

        string packageName;

        float packagePrice;



public:

        MembershipPackage() {};

        MembershipPackage(string Id, string name, float price) {

                membershipId = Id;

                packageName = name;

                packagePrice = price;

        }

        void displayMembershipPackage() {

                cout << "Membersip Id : " << membershipId << endl;

                //cout << "Membersip Name : " << packageName << endl;

                //cout << "Membersip Price : " << packagePrice << endl;

        }

        ~MembershipPackage() {

                cout << "Deleting Membersip " << membershipId << endl;

        }

};


class Customer {

private:

        string customerId;

        string customerName;
```

```
	int age;
	string gender;
	string email;
	UserAccount *account;

public:
	Customer(string cusId, string name, int cusAge, string cusGender, string pemail, string uName) {
		account = new UserAccount(uName);
		customerId = cusId;
		customerName = name;
		age = cusAge;
		email = pemail;
	}
	void DisplayUserAccounts() {
		//for (int r = 0; r < SIZE; r++)
			account->Display();
	}
	string getCustomer() {
		return customerName;

	}
	void displayCustomer() {
		cout << "Customer ID : " << customerId << endl;
	}
	~Customer() {
		cout << "Customer deleted " << endl;
		//for (int r = 0; r < SIZE; r++)
			delete account;
		cout << "Everthing is deleted" << endl;
	}
};

class Subscription
{
private:
	Customer *cusSub;
	MembershipPackage *pkg;
	string date;
public:
	Subscription(Customer *pcusSub, MembershipPackage *ppkg, string pdate)
	{
		cusSub = pcusSub;
		pkg = ppkg;
		date = pdate;
	}
	void displaySubscription()
	{
		cusSub->displayCustomer();
		pkg->displayMembershipPackage();
		cout << "Date =" << date << endl;
```

```
        }
        ~Subscription() {
                cout << "\nDeleting Subscription " << endl;
        }
};

class Report {


private:
        string name;
public:
        void getReport(Customer *c)
        {
                name = c->getCustomer();
        }
        void display()
        {
                cout << "Customer name = " << name << endl;
        }
};

int main()
{
        // User Account
        Customer *cus;
        cus = new Customer("C0001", "Kamal", 25, "Male", "123@gmail.com", "U0001");
        cus->DisplayUserAccounts();

        //delete cus;

        cout << "***************\n" << endl;


        // subscription

        MembershipPackage *P = new MembershipPackage("P001", "Standard", 10000.00);

        Subscription subs1(cus, P, "20/05/2022");

        subs1.displaySubscription();

        cout << "***************\n" << endl;


        // report

        Report *R = new Report();

        R->getReport(cus);

        R->display();

        cout << "***************\n" << endl;


        return 0;

}
```

***

--- IT21168222 ---

--- IT21170348 ---

```
#include<iostream>

using namespace std;


//Payment class


class Payment{

 private:

 int Payment_ID;

 string cus_ID;

 float amount;


 public:

 Payment();

 Payment (int Pay_ID,string c_ID,float Amount);

 void Paymentdetailes();

 void validate();

 };




Payment :: Payment(){}


Payment :: Payment(int Pay_ID,string c_ID,float Amount) {

 int Payment_ID = Pay_ID;

 string Cus_ID = c_ID;

 float amount = Amount;
```

```
}
void Payment:: Paymentdetailes() { }
void Payment:: validate () { }



 //MembershipPackage class
class MembershipPackage{

 private:
            string package_id;
            char package_Name;
            float package_Price;
            string session_Time;

 public:
            MembershipPackage();
            MembershipPackage(string p_id,char p_name,float p_price,string s_time);
            void MembershipPackagedetailes();
};

MembershipPackage :: MembershipPackage(string p_id,char p_name,float p_price,string
s_time){
            string package_id = p_id;
            char package_Name = p_name;
            float package_Price = p_price;
            string session_Time = s_time;
```

```
}

void MembershipPackage :: MembershipPackagedetailes(){

}


//Item class

class Item{

        private:

                string name;

                string itemid;

                double price;

                int quantity;

                Payment * pay;


        public:

                Item();

                Item(string pname,string pitemid,double pprice,int pquantity);

                void additem();

                void Quantity();

                void calcPrice();

};


Item :: Item(){}

Item :: Item(string pname,string pitemid,double pprice,int pquantity){


      name=pname;

      itemid=pitemid;
```

```
          price=pprice;

          quantity=pquantity;

}


void Item :: additem(){}

void Item :: Quantity(){}

void Item :: calcPrice(){}




//report class

class report{

 private:

            string reportName;

            int reportNo;


 public:

            report();

            report (string r_name,int r_no);

            void displayreport();

};


report :: report(string r_name,int r_no){

            string reportName = r_name;

            int reportNO= r_no;


}
```

```
void report :: displayreport(){
}

//Cart class
class cart{
        private:
                int cardID;
                char product;
                double price;
                double total;

        public:
                cart();
                cart (int cartID,char p,double pprice,double t);
                void displaytotalprice();
};

cart :: cart(int c_id,char p,double pprice,double t){
                int cardID= c_id;
                char product= p;
                double price= pprice;
                double total= t;
}
void cart :: displaytotalprice(){
}
```

```
int main(){
Payment *Pay1,*Pay2;
Pay1= new Payment;
Pay2= new Payment;

delete Pay1,Pay2;

return 0;
}
```

***

-- IT21170348 --

```
float totalamount;
class item
{
private:
    char name[25];
    int qty;
    float rate,amount;
public:
    void getdata()
    {
            cout<<endl<<"Enter the item name:";
            cin>>name;
            cout<<"\nEnter the rate:";
            cin>>rate;
            cout<<"Entet the quantity:";
            cin>>qty;
    }
    void printdata()
    {
            cout<<setw(12)<<name;
            cout<<setw(9)<<qty;
            cout<<setw(9)<<rate;
            amount=rate*qty;
            totalamount=amount+totalamount;
            cout<<setw(8)<<amount<<endl;
    }
};
void main()
{
    item shop[10];
    char ch='y';
    int n=0,i;
    clrscr();
    while (ch=='y'||ch=='y')
    {
            shop[n].getdata();
            n++;
            cout<<"do you want to countinue u\n"<<endl;
            cin>>ch
    }
    count<<"\t\t*********"<<endl;
    cout<<"\t\t shoping list"<<endl;
    cout<<"\t**************************"<<endl;
    cout<<name quantity-rate-amount\n";
            for(i=0;i<n;i++)
                    shop[i].printdata();
    cout<<"\t-------------------------------"<<endl;
    cout<<"\ttotal bill:"<<totalamount;
    getch();
}
return 0;


class feedback
{
private:
int feedbackId;
char user Name[20];
char feedback Messsage[300];
user*user1;
public:
feedback();
feedback(int fbIp,char user Name[],char desc[]);
void displayfeedback();
};
feedback::feedback(){
feedbackId=0;
strcpy(userName,"");
strcpy(feedback Message,"");
}
feedback::feedback(intfbId,char user Name[],char desc[]){
feedbackId=fbid;
strcpy(user Name);
strcpy(feedback Message,desc);
}
void feedback::displayfeedback(){}
return 0;
}
```

***

---IT21171710--

```
#include <iostream>
#include <cstring>
#include <string>
using namespace std;
#define SIZE 2;

//Trainer class

class Trainer{
  private:
     Session * strn[SIZE];
     string trainerid[10];
     string trainerName[20];
     string trainerEmail[20];
     double salary[20];

  public:
    Trainer();
    Trainer(string ptrainerid,string ptrainerName[],string ptrainerEmail[],double psalary[]);
    void gettrainerDetails();
    void showtrainerDetails();
    void displayDetails(Session * strn1, Session * strn2 );

};
```

```
// Session class

class Session{
  private:
   string sessionName[10];
   string sessionid[10];

  public:
   Session();
   Session(string psessionName[],string psessionid);
   void displayDetails;

};

// MembershipPackage class

class MembershipPackage{
      private:
            string packageid[];
            string packageName[];
            float packagePrice;
            string sessiontime[];
            Session * package[SIZE];


  public:
      MembershipPackage();
```

```
    MembershipPackage(string ppackageid,string ppackageName[],float ppackgrPrice,string psessiontime[]);

    void packageDetails();

    void sessiontime();




};


// Payment class


class Payment{

    private:

        string paymentid[];

        string paymentMethod[];

        string customerid[];

        float amount;

        Item * ite[SIZE];


  public:

    Payment();

    Payment(string ppaymentid,string ppaymentMethod[],string pcustomerid[],float pamount);

    void paymentDetails(Item * tem);

    void validate();




};
```

```
// Item class


class Item{
        private:
                string name[];
                string itemid[];
                double price;
                int quantity;
                Payment * pay;


        public:
                Item();
                Item(string pname[],string pitemid[],double pprice,int pquantity,Payment *
ppay);
                void additem();
                void Quantity();
                void calcPrice();
};


//  Session class implementation




Session :: Session() {}
Session :: Session(string psessionName,string psessionid){
        sessionName=psessionName;
                        sessionid=psessionid;
                        }
```

```
void Session :: displayDetails(){}

~Sesiion(){

        cout<<"Deleting session"<< sessionid << endl;

}




// Trainer class  implementation


Trainer :: Trainer() {}

Trainer :: Trainer(string ptrainerid,string ptrainerName,string ptrainerEmail,double psalary){

             trainerid=ptrainerid;

             trainerName=ptrainerName;

             treainerEmail=ptrainerEmail;

             salary=psalary;

}


void Trainer :: displayDetails(Session * strn1,Session * strn2)

{

        strn[0]=strn1;

        strn2[1]=strn2;

}


void Trainer :: gettrainerDetails() {

         cout<<"\nEnter the trainer ID: "  <<endl;

         cin trainerid;

         cout<<"\nEnter trainer Name: " <<endl;
        cin<<trainerName;
        cout<<"\nEnter trainer Email: "  <<endl;
        cin<< trainerEmail;
        cout<<"\n Enter trainer salary amount: " <<endl;
        cin<<salary;

void Trainer :: showtrainerDetails() {}

// MembershipPackage class implementation

MembershipPackage :: MembershipPackage() {}
MembershipPackage :: MembershipPackage(string ppackageid,string ppackageName,float ppackgrPrice,string psessiontime){

        packageid=ppackageid;
        packageName=ppackageName;
        packagePrice=ppackagePrice;
        sessiontime=psessiotime;
}

void MembershipPackage :: packageDetails (){}
void MembershipPackage :: sessiontime (){}

~MembershipPackage() {cout << "session over" <, ensl;
    for(int i=0; i<[SIZE]);      i++)
       delete session[i];
   cout << "End" << endl;
```

```
}


// Payment class implementation


Payment :: Payment() {}

Payment :: Payment(string ppaymentid,string ppaymentMethod,string pcustomerid,float pamount){


        paymentid=ppaymentid;

        paymentMethod=ppaymentMethod;

        customerid=pcustomerid;

        amount=pamount;

}


void Payment :: paymentDetails (){}

void Payment :: validate () {}


//Item class implementation


Item :: Item(){}

Item ::Item(string pname,string pitemid,double pprice,int pquantity){


        name=pname;

        itemid=pitemid;

        price=pprice;

        quantity=pquantity;

}
```

```
void Item :: additem(){}

void Item :: Quantity(){}

void Item :: calcPrice(){}


// main program


int main(){

	Trainer tr("T001","Henry","henry456@gmail.com","Rs.100 000.00");


	return 0;

}
```

***

-- IT21171710--

--IT21171406--

```
#include<iostream>

#include<string>

 using namespace std;

//Dependancy(Trainer and Report)

class Trainer

{

    private:
```

```
  string trainerid;

  string trainerName;

  string trainerEmail;

  double salary;


   public:

   Trainer();

 void getTrainerTrainerDetails(){


            string gettrainerid(){


   return trainerid;

}

string gettrainerName(){




   return trainerName;

}

    string gettrainerEmail(){


   return trainerEmail;

}

   double getsalary(){


   return salary;


 void showTrainerDetails(){
}


count<<"trainerid="trainerid<<endl;
count<<"trainername="trainername<<endl;
count<<"trainerEmail="trainerEmail<<endl;
count<<"salary="salary<<endl;

}

class Report
{
private:
 string reportName;
 int reportNo;



public:
 Report(string reportName){

  reportName=TrainersReport;
}
void display()
{
count<<"reportname="reportname<<endl;
count<<"reprtNo=   "reportNo<<endl;
}
};
```

```
//Dependeny(MembershipPackage and report)

class MembershipPackage

{
        private:
                string packageid;
                string packageName;
                float  packagePrice;
                string sessiontime;

        public:
            void packageDetails(){
                count<<"packageid="packageid<<endl;
        count<<"packagename="packagename<<endl;
        count<<"packagePrice="packagePrice<<endl;
        count<<"sessiontime="sessiontime<<endl;

                 }


};



class Report
```

```
{
private:
 string reportName;
 int reportNo;



public:
 Report(string reportName){
            reportName=MembershipPackagesReport;
 }

void display()
{
count<<"reportname="reportname<<endl;
count<<"reportNo=   "reportNo<<endl;
}
};




 //Aggregation(Trainer and MembershipPackage)
 class Trainer
{
    private:
  string trainerid;
  string trainerName;
  string trainerEmail;
  double salary;

   public:
Trainer();
 void getTrainerTrainerDetails(){
   return trainerid;
   return trainerName;
   return trainerEmail;
   return salary;

 void showTrainerDetails(){

count<<"trainerid="trainerid<<endl;
count<<"trainername="trainername<<endl;
count<<"trainerEmail="trainerEmail<<endl;
count<<"salary="salary<<endl;

Trainer() {count<<"Deleting Trainer"<<Trainerid<<endl;
}




 class MembershipPackage

 {
           private:
            Trainer*trainer[2];

            string packageid;

            string packageName;

            float  packagePrice;

            string sessiontime;


public:

            MembershipPackage(){};


            void addTrainer(Trainer*trainer1,Trainer*trainer2)

            {

                        trainer[0]=trainer1;

                        trainer[1]=trainer2;

             }


     void packageDetails(){

   count<<"packageid="packageid<<endl;

count<<"packagename="packagename<<endl;

count<<"packagePrice="packagePrice<<endl;

count<<"sessiontime="sessiontime<<endl;


             }


  void displayMembershipPackage(){

                        for(int i=0; i<SIZE;i++)

                        trainer[i]->displayTrainer();

             }
```

```
int main(){
        Trainer*t1= new Trainer ("T001","Kularathne");
        Trainer*t2= new Trainer ("T002","Rathnayake");

        addTrainer(t1,t2);
        delete t1;
        delete t2;

        Trainer*t3= new Trainer ("T003","Ekanayake");
        addTrainer(t3);
        displayMembershipPackage();

        return 0;
}
```

***

--IT21171406---

Individual Contribution

	Student ID	Student Name	Individual Contribution
1	IT21168222	Saranga G K G	• Created the CRC Card for Customer, UserAccount and Subscription classes. • Draw Customer, UserAccount and Subscription classes in the class diagram. • Implemented the coding for o the Customer class and the UserAccount class o the Customer class and the MembershipPackage class o the Customer class and the Report class
2	IT21170348	Dinujaya K G M	• created CRC card for payment and membershipPackage . • Draw payment and membershipPackage classes in the class diagram. • Coded parts: ▪ Payment class and MembershipPackage class. ▪ The Item class and the report class. ▪ The cart class and item class.
3	IT21168604	Weihena K M	• Draw CRC cards for cart class and feedback class diagram. • Draw feedback,cart class in the diagram • Coded parts: 1. customer class and the cart class 2. The feedback class & the customer
4	IT21171710	Oshadari K M R P	• Draw CRC cards for Trainer class and Item class. • Draw Trainer class and Item class in the class diagram. • Coded parts: 1. The Trainer class and the Session class 2. The MembershipPackage and the session class 3. The Item class and the Payment class
5	IT21171406	Ekanayake E P M T S	• Draw CRC cards for Session class and Report class. • Draw Session and Report class in the class diagram • Coded parts: 1. Report class and Membership Package class 2. Trainer class and Report class 3. Trainer class and Membership Package class