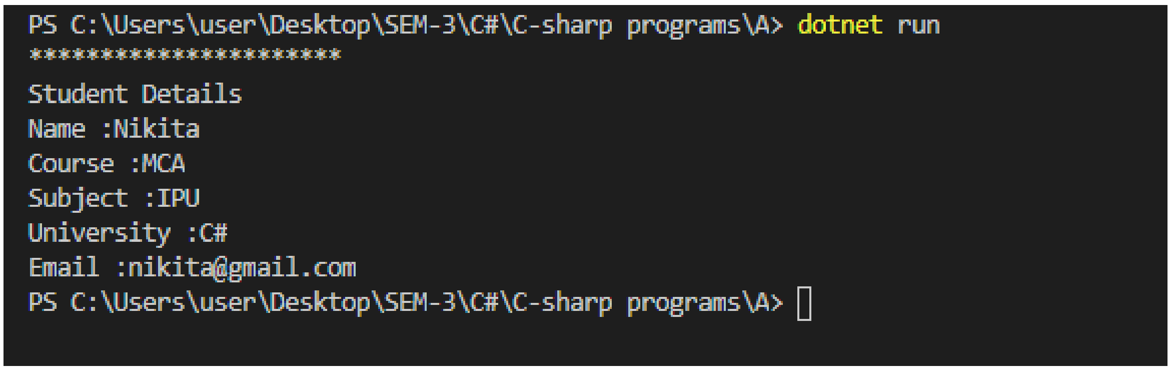
NIKITA KAPOOR
01435304419

Lab Assignment # 4

AP2 : Declare several constructors for the class Student, which have different lists of parameters (for complete information about a student or part of it). Data, which has no initial value to be initialized with null. Use nullable types for all nonmandatory data.

```
using System;

namespace LAB2_24_7
{
    class Student
    {
        public string full_name;
        public string course;
        public string subject;
        public string university;
        public string email;
        public long? phone_number;
        public Student(){}
        public Student(string full_name,string course,string university,string subject,string email,int phone_number){
            this.full_name=full_name;
            this.university=university;
            this.course=course;
            this.subject=subject;
            this.email=email;
            this.phone_number=phone_number;
        }
        public Student(string full_name,string course,string university){
                this.full_name=full_name;
                this.course=course;
                this.university=university;
                phone_number=null;
                email=null;
                subject=null;
        }
         public Student(string full_name,string course,string university,string subject,string email){
            this.full_name=full_name;
            this.university=university;
            this.course=course;
            this.subject=subject;
            this.email=email;
        }
        public void getDetails(){
            Console.WriteLine("*********************");
            Console.WriteLine("Student Details");
            Console.WriteLine("Name :"+full_name);
            Console.WriteLine("Course :"+course);
            Console.WriteLine("Subject :"+subject);
            Console.WriteLine("University :"+university);
            Console.WriteLine("Email :"+email);
            if(phone_number!=null)
                Console.WriteLine("Phone Number :"+phone_number);
        }
        public void setDetails(){
            Console.WriteLine("Enter Name");
            full_name = Console.ReadLine();
            Console.WriteLine("Enter Course");
            course = Console.ReadLine();
            Console.WriteLine("Enter Subject");
            subject = Console.ReadLine();
            Console.WriteLine("Enter University");
            university = Console.ReadLine();
            Console.WriteLine("Enter Email");
            email = Console.ReadLine();
            Console.WriteLine("Enter Phone Number");
            phone_number = Convert.ToInt64(Console.ReadLine());

        }

    }
    class Program
    {
        static void Main(string[] args)
        {
            Student s=new Student("Nikita","MCA","C#","IPU","nikita@gmail.com");
            //s.setDetails();
            s.getDetails();

        }
    }
}
```

OUTPUT



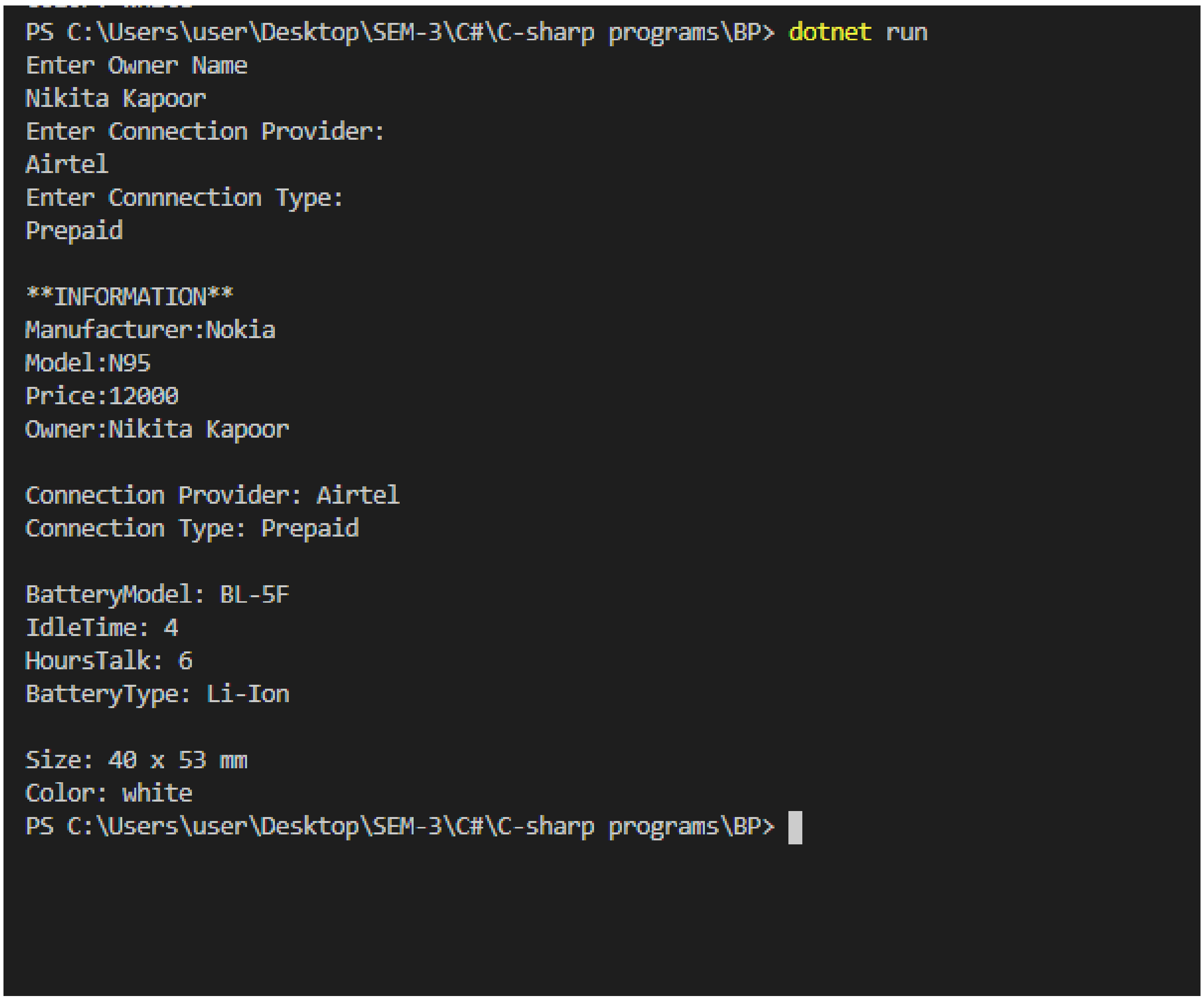
```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\A> dotnet run
**********************
Student Details
Name :Nikita
Course :MCA
Subject :IPU
University :C#
Email :nikita@gmail.com
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\A>
```

NIKITA KAPOOR
01435304419

BP2 Declare several constructors for each of the classes created by the previous task, which have different lists of parameters (for complete information about a student or part of it). Data fields that are unknown have to be initialized respectively with null or 0

```
using System;

namespace BP
{
    class Mobile{
        public string model;
        public string manufacturer;
        public double price;
        public string owner;
        public Mobile(){
            model=null;
            price=0;
            manufacturer=null;
            owner=null;
        }
        public Mobile(string model,string manufacturer,double price ,string owner){
            this.model=model;
            this.manufacturer=manufacturer;
            this.price=price;
            this.owner=owner;
        }
        public static string[] NokiaN95={"Nokia","N95","12000","BL-
5F","4","6","1","40 x 53 mm","white"};
        public void StoreGeneralInformation(){
                    Console.WriteLine("Enter Model:");
                    model=Console.ReadLine();
                    Console.WriteLine("Enter Manufacturer:");
                    manufacturer=Console.ReadLine();
                    Console.WriteLine("Enter Price:") ;
                    price=Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("Enter Owner Name:");
                    owner=Console.ReadLine();
        }
        public void StoreOwnerInfo(){
           Console.WriteLine("Enter Owner Name");
           owner=Console.ReadLine();
        }
        public string MobileInfo(){
             return ("Manufacturer:"+manufacturer+"\nModel:"+model+"\nPrice:"+price+"\nOwn
er:"+owner);
        }
         public void NokiaInfo(){
            manufacturer=NokiaN95[0];
            model=NokiaN95[1];
            price=Convert.ToInt32(NokiaN95[2]);
        }
    }
    class GSM:Mobile{
        string connection_Provider;         //BSNL, AIRTEL, IDEA, JIO
        string connection_type;             //PREPAID, POSTPAID

        public  Battery battery;
        public Screen screen;
        public void StoreGSMInformation(){
                    Console.WriteLine("Enter Connection Provider:");
                    connection_Provider=Console.ReadLine();
                    Console.WriteLine("Enter Connnection Type:");
                    connection_type=Console.ReadLine();
        }
        public string NokiaDisplayInfo(){
            NokiaInfo();
            battery=new Battery(NokiaN95[3],Convert.ToInt32(NokiaN95[4]),Convert.ToInt32(
NokiaN95[5]));
            screen=new Screen(NokiaN95[7],NokiaN95[8]);
            StoreOwnerInfo();
            StoreGSMInformation();
            Console.WriteLine("\n**INFORMATION**");
            string infoAboutPhone = MobileInfo()+"\n"+"\nConnection Provider: "+connection_P
rovider+
            "\nConnection Type: "+connection_type+"\n\n"+battery.GetInformationBattery() +
            "\nBatteryType: "+battery.GetBatteryType()+ "\n\n"+
            screen.GetInformationScreen() ;
            return infoAboutPhone;
        }
    }
    class Battery{
        public string batteryModel;
        public int idle_time;
        public int hours_talk;

        public enum BatteryType{LiIon=1,NiMH,NiCd};
        public BatteryType batteryType=(BatteryType)1;
        public Battery(){
            batteryModel=null;
            idle_time=0;
            hours_talk=0;
        }

        public Battery(string batteryModel,int idle_time, int hours_talk){
            this.batteryModel=batteryModel;
            this.idle_time=idle_time;
            this.hours_talk=hours_talk;
        }
        public void StoreInformationBattery(){
                    Console.WriteLine("Enter Battery Model:");
                    batteryModel=Console.ReadLine();
                    Console.WriteLine("Enter Idle Time:") ;
                    idle_time=Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("Enter Hours Talk:") ;
                    hours_talk=Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("Enter Choice for Battery Type:") ;
                     Console.WriteLine("1.Li-Ion\n2.NiMH\n3.Nicd") ;
                    batteryType=(BatteryType)Convert.ToInt32(Console.ReadLine());
        }
        public string GetInformationBattery(){
                    return("BatteryModel: "+batteryModel+"\nIdleTime: "+idle_time+"\nHoursTalk:
"+hours_talk);
        }
        public string GetBatteryType()
        {
            switch (batteryType)
            {
                case BatteryType.LiIon:
                    return "Li-Ion";
                case BatteryType.NiMH:
                    return "NiMH";
                case BatteryType.NiCd:
                    return "NiCd";
                default:
                    return ("Unsupported battery type: " + batteryType);
            }
        }
    }
    class Screen{
      public string size;
      public string color;
      public Screen(){
          size=null;
          color=null;
      }
      public Screen(string size,string color){
          this.size=size;
          this.color=color;
      }
      public void StoreInformationScreen(){
                    Console.WriteLine("Enter Size:");
                    size=Console.ReadLine();
                    Console.WriteLine("Enter Color:") ;
                    color=Console.ReadLine();
        }
         public string GetInformationScreen(){
                return("Size: "+size+"\nColor: "+color);
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            GSM gsm=new GSM();
            Console.WriteLine(gsm.NokiaDisplayInfo());
        }
    }
}
```

OUTPUT



```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\BP> dotnet run
Enter Owner Name
Nikita Kapoor
Enter Connection Provider:
Airtel
Enter Connnection Type:
Prepaid

**INFORMATION**
Manufacturer:Nokia
Model:N95
Price:12000
Owner:Nikita Kapoor

Connection Provider: Airtel
Connection Type: Prepaid

BatteryModel: BL-5F
IdleTime: 4
HoursTalk: 6
BatteryType: Li-Ion

Size: 40 x 53 mm
Color: white
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\BP>
```

NIKITA KAPOOR
01435304419

CP1 Create a class Call, which contains information about a call made via mobile phone. It should contain information about date, time of start and duration of the call.

CP2 Add a property for keeping a call history – CallHistory, which holds a list of call records.

CP3 In GSM class add methods for adding and deleting calls (Call) in the archive of mobile phone calls. Add method, which deletes all calls from the archive.

CP4 In GSM class, add a method that calculates the total amount of calls (Call) from the archive of phone calls (CallHistory), as the price of a phone call is passed as a parameter to the method.

```
using System;

namespace BP
{
    class Mobile{
        public string model;
        public string manufacturer;
        public double price;
        public string owner;
        public Mobile(){
            model=null;
            price=0;
            manufacturer=null;
            owner=null;
        }
        public Mobile(string model,string manufacturer,double price ,string owner){
            this.model=model;
            this.manufacturer=manufacturer;
            this.price=price;
            this.owner=owner;
        }
        public static string[] NokiaN95={"Nokia","N95","12000","BL-
5F","4","6","1","40 x 53 mm","white"};
        public void StoreGeneralInformation(){
                    Console.WriteLine("Enter Model:");
                    model=Console.ReadLine();
                    Console.WriteLine("Enter Manufacturer:");
                    manufacturer=Console.ReadLine();
                    Console.WriteLine("Enter Price:") ;
                    price=Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("Enter Owner Name:");
                    owner=Console.ReadLine();
        }
        public void StoreOwnerInfo(){
            Console.WriteLine("Enter Owner Name");
            owner=Console.ReadLine();
        }
        public string MobileInfo(){
             return ("Manufacturer:"+manufacturer+"\nModel:"+model+"\nPrice:"+price+"\nOwn
er:"+owner);
        }
         public void NokiaInfo(){
            manufacturer=NokiaN95[0];
            model=NokiaN95[1];
            price=Convert.ToInt32(NokiaN95[2]);
        }
    }
    class GSM:Mobile{
        string connection_Provider;         //BSNL, AIRTEL, IDEA, JIO
        string connection_type;             //PREPAID, POSTPAID

        public  Battery battery;
        public Screen screen;
        static int counter=0;
        public Call[] call=new Call[500];
        public void TotCost(double price){
            double sum=0;
             for(int i=0;i<counter;i++){
                sum+=(Convert.ToInt32(call[i].CallHistory[2])*price);
            }
            Console.WriteLine("Total Price: "+sum+" Rs");
        }
        public void AddCalls(){
            Console.WriteLine("Enter Date:(eg.21-08-2020)");
            string date=Console.ReadLine();
            Console.WriteLine("Enter StartTime:(eg:14:05)");
            string startTime=Console.ReadLine();
            Console.WriteLine("Enter Duration: (seconds)");
            string duration=Console.ReadLine();
            if(Call.totCalls<500){
                call[counter]=new Call();
                call[counter].CallHistory[0]=date;    //CallHistory is property
                call[counter].CallHistory[1]=startTime;
                call[counter].CallHistory[2]=duration;
                Call.totCalls++;
                counter++;
                Console.WriteLine("Call Record Added");
            }
            else
                Console.WriteLine("Call Record Full");
        }
         public void showCalls(){
            for(int i=0;i<counter;i++){
                for(int j=0;j<3;j++){
                    Console.WriteLine(call[i].CallHistory[j]);
                }
            }
        }
        public void DeleteAllCalls(){
            for(int i=0;i<counter;i++){
                  for(int j=0;j<3;j++){
                   call[i].CallHistory[j]=null; // delete all calls
                }
            }
             Console.WriteLine("All Call Records Deleted");
        }
         public void DeleteCalls(string date,string time){
             int flag=0;
              for(int i=0;i<counter;i++){
                   if( call[i].CallHistory[0].Equals(date)&&  call[i].CallHistory[1].Equals(time)){
                       flag=1;
                       for(int j=0;j<3;j++){
                           call[i].CallHistory[j]=null; // delete particular calls
                        }
                   }
                }
                Console.WriteLine((flag==0) ? "Call Record Not Found" : "Call Record Deleted");
        }
        public void StoreGSMInformation(){
                    Console.WriteLine("Enter Connection Provider:");
                    connection_Provider=Console.ReadLine();
                    Console.WriteLine("Enter Connnection Type:");
                    connection_type=Console.ReadLine();
        }
        public string NokiaDisplayInfo(){
            NokiaInfo();
            battery=new Battery(NokiaN95[3],Convert.ToInt32(NokiaN95[4]),Convert.ToInt32(
NokiaN95[5]));
            screen=new Screen(NokiaN95[7],NokiaN95[8]);
            StoreOwnerInfo();
            StoreGSMInformation();
            Console.WriteLine("\n**INFORMATION**");
            string infoAboutPhone = MobileInfo()+"\n"+"\nConnection Provider: "+connection_P
rovider+
            "\nConnection Type: "+connection_type+"\n\n"+battery.GetInformationBattery() +
            "\nBatteryType: "+battery.GetBatteryType()+ "\n\n"+
            screen.GetInformationScreen() ;
            return infoAboutPhone;
        }
    }
```

```
 class Battery{
    public string batteryModel;
    public int idle_time;
    public int hours_talk;

    public enum BatteryType{LiIon=1,NiMH,NiCd};
    public BatteryType batteryType=(BatteryType)1;
    public Battery(){
        batteryModel=null;
        idle_time=0;
        hours_talk=0;
    }

    public Battery(string batteryModel,int idle_time, int hours_talk){
        this.batteryModel=batteryModel;
        this.idle_time=idle_time;
        this.hours_talk=hours_talk;
    }
    public void StoreInformationBattery(){
                Console.WriteLine("Enter Battery Model:");
                batteryModel=Console.ReadLine();
                Console.WriteLine("Enter Idle Time:") ;
                idle_time=Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Enter Hours Talk:") ;
                hours_talk=Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Enter Choice for Battery Type:") ;
                 Console.WriteLine("1.Li-Ion\n2.NiMH\n3.Nicd") ;
                batteryType=(BatteryType)Convert.ToInt32(Console.ReadLine());
    }
    public string GetInformationBattery(){
                return("BatteryModel: "+batteryModel+"\nIdleTime: "+idle_time+"\nHoursTalk: "+hours_talk);
    }
    public string GetBatteryType()
    {
        switch (batteryType)
        {
            case BatteryType.LiIon:
                return "Li-Ion";
            case BatteryType.NiMH:
                return "NiMH";
            case BatteryType.NiCd:
                return "NiCd";
            default:
                return ("Unsupported battery type: " + batteryType);
        }
    }
 }
```

NIKITA KAPOOR
01435304419

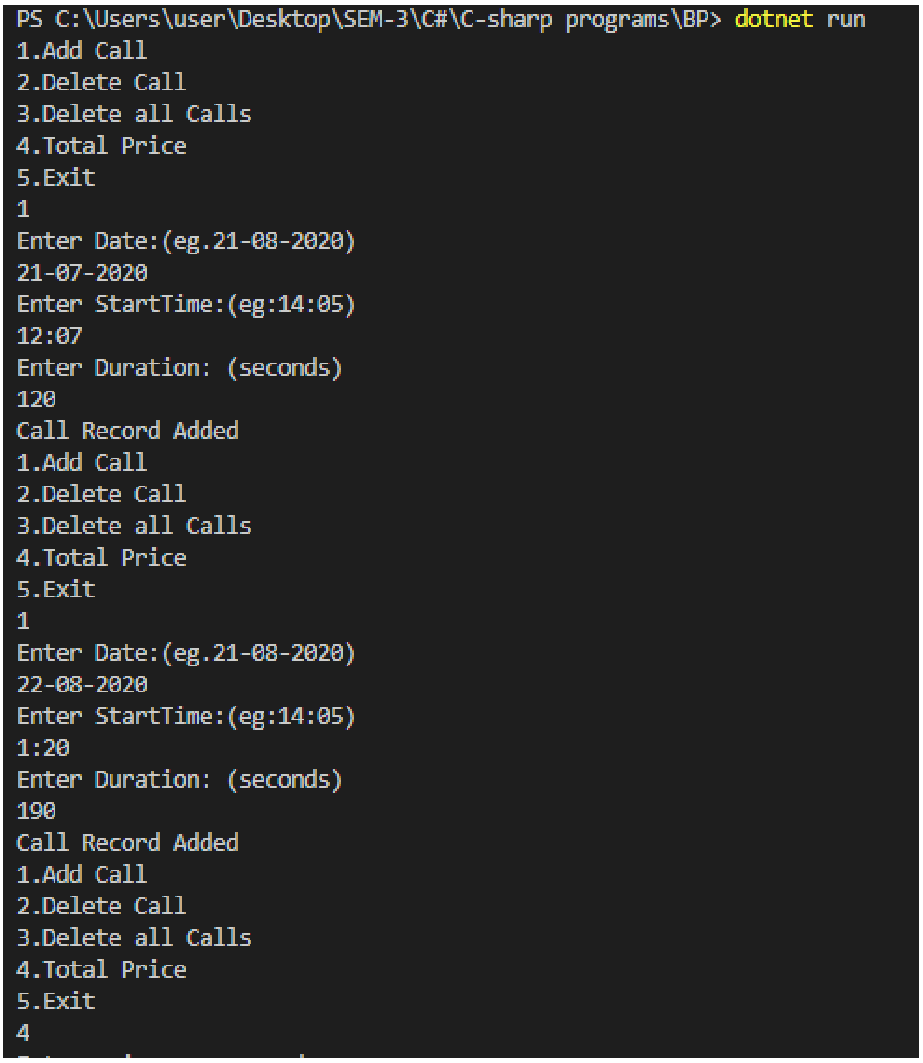
```
class Screen{
   public string size;
   public string color;
   public Screen(){
       size=null;
       color=null;
   }
   public Screen(string size,string color){
       this.size=size;
       this.color=color;
   }
   public void StoreInformationScreen(){
               Console.WriteLine("Enter Size:");
               size=Console.ReadLine();
               Console.WriteLine("Enter Color:") ;
               color=Console.ReadLine();
    }
     public string GetInformationScreen(){
               return("Size: "+size+"\nColor: "+color);
    }
}
class Call
{
    string date;
    string startTime;
    int duration;
    public static int totCalls=0;
    string[] callHistory=new string[3];
    public Call(){}
    public Call(string date,string startTime,int duration){
        this.date=date;
        this.startTime=startTime;
        this.duration=duration;
    }
    public  string[] CallHistory{
            get
            {
                return callHistory;
            }
            set
            {
                callHistory=value;
            }
    }
     public string GetInformationCall(){
            return("Date: "+date+"\nStartTime: "+startTime+"\nDuration: "+duration);
    }
}
```

NIKITA KAPOOR
01435304419

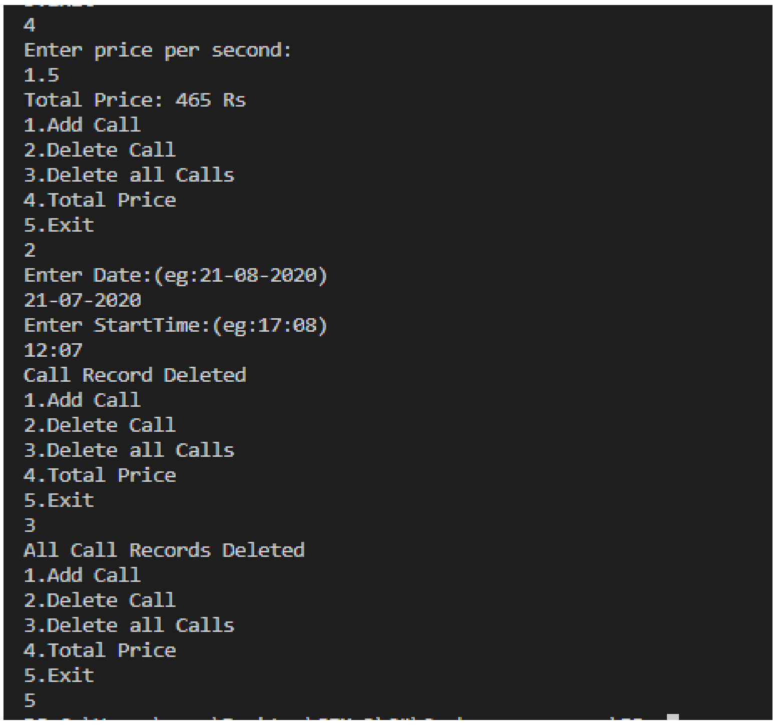
```
    class Program
    {
        static void Main(string[] args)
        {
            int ch;
            GSM gsm=new GSM();
            do{
                Console.WriteLine("1.Add Call");
                Console.WriteLine("2.Delete Call");
                Console.WriteLine("3.Delete all Calls");
                Console.WriteLine("4.Total Price");
                Console.WriteLine("5.Exit");
                ch=Convert.ToInt32(Console.ReadLine());
                switch(ch){
                    case 1: gsm.AddCalls(); break;
                    case 2: Console.WriteLine("Enter Date:(eg:21-08-2020)");
                            string date=Console.ReadLine();
                            Console.WriteLine("Enter StartTime:(eg:17:08)");
                            string time=Console.ReadLine();
                            gsm.DeleteCalls(date,time);
                            break;
                    case 3: gsm.DeleteAllCalls(); break;
                    case 4: Console.WriteLine("Enter price per second:");
                            double price=Convert.ToDouble(Console.ReadLine());
                            gsm.TotCost(price);
                            break;
                }
            }
            while(ch!=5);
        }
    }
}
```

NIKITA KAPOOR
01435304419

OUTPUT



```
PS C:\Users\user\Desktop\SEM-3\C#\C-sharp programs\BP> dotnet run
1.Add Call
2.Delete Call
3.Delete all Calls
4.Total Price
5.Exit
1
Enter Date:(eg.21-08-2020)
21-07-2020
Enter StartTime:(eg:14:05)
12:07
Enter Duration: (seconds)
120
Call Record Added
1.Add Call
2.Delete Call
3.Delete all Calls
4.Total Price
5.Exit
1
Enter Date:(eg.21-08-2020)
22-08-2020
Enter StartTime:(eg:14:05)
1:20
Enter Duration: (seconds)
190
Call Record Added
1.Add Call
2.Delete Call
3.Delete all Calls
4.Total Price
5.Exit
4
```



```
4
Enter price per second:
1.5
Total Price: 465 Rs
1.Add Call
2.Delete Call
3.Delete all Calls
4.Total Price
5.Exit
2
Enter Date:(eg:21-08-2020)
21-07-2020
Enter StartTime:(eg:17:08)
12:07
Call Record Deleted
1.Add Call
2.Delete Call
3.Delete all Calls
4.Total Price
5.Exit
3
All Call Records Deleted
1.Add Call
2.Delete Call
3.Delete all Calls
4.Total Price
5.Exit
5
```