**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();

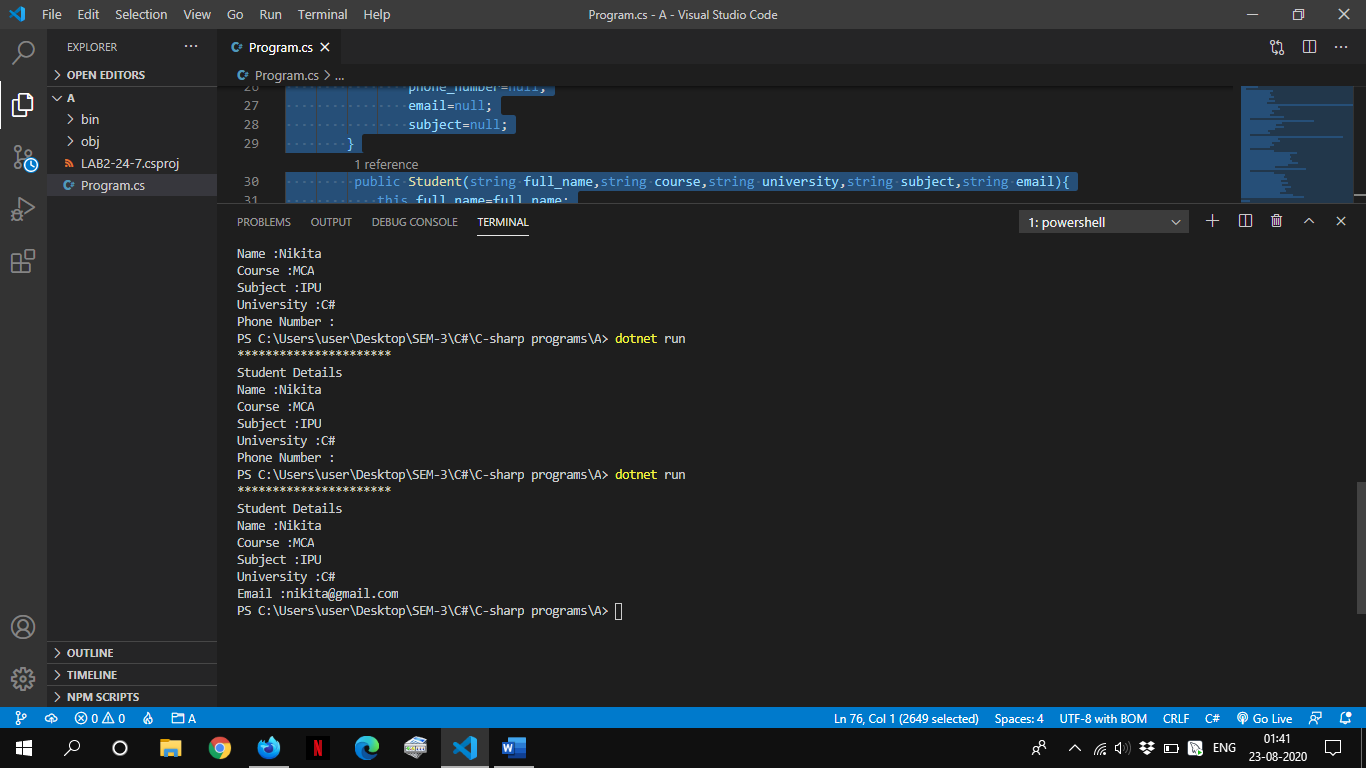
            s.getDetails();

        }

    }

}

**OUTPUT**



**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+"\nOwner:"+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\_Provider+

            "\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();

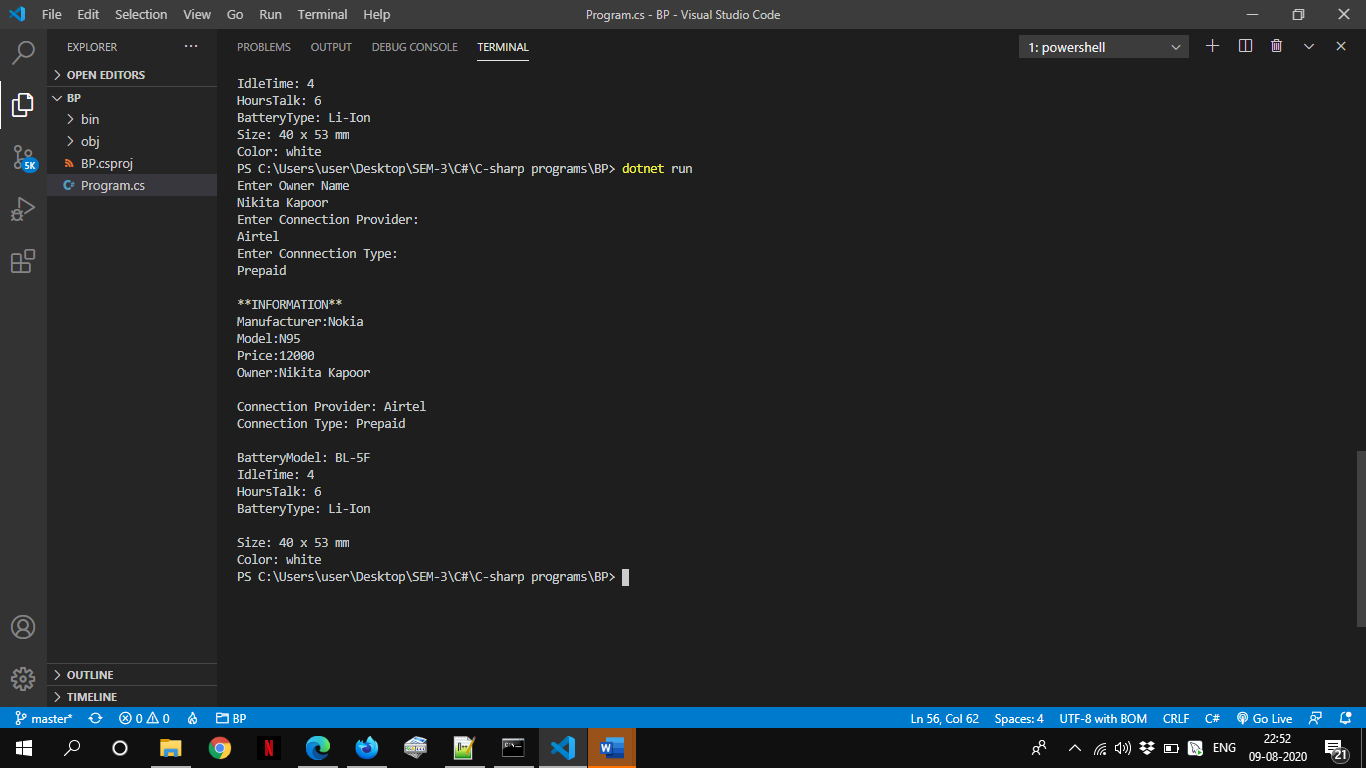
            Console.WriteLine(gsm.NokiaDisplayInfo());

        }

    }

}

**OUTPUT**



**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+"\nOwner:"+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\_Provider+

            "\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 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);

        }

    }

    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);

        }

    }

}

**OUTPUT**

