Assignment
By
Triveni Anumolu

AMAZON Class Product:

Code:

```
class Products
    {
      public int price;
      public string brand;
      public string quantity;
      public string colour;

      public static void Addproductprice()
      {
        //TODO
      }
       public static void Editproductprice()
      {
        //TODO
      }
      public static void Deleteproductbrand()

      {
        //TODO

      }
      public static void Displayproductcolour()

      {
        //TODO
      }
    }
```

UML :

Products
+product price   : int +product brand   : string +product quantity   : string +product colour   : string

+Addproductprice () : void();
+Editproduct price   () : void();
+Deleteproductcolour () : void();
+Displayproductcolour (): void();

AMAZON Class Employees:

Code:

```
class Employee
        {
            Public string emp name;
            public string emp id;
            private int empsalry;
            public string empdesignation;

                public static void Addemployeeid()
            {
                //TODO
            }
             public static void Editemployeeid()
            {
                //TODO
            }
            public static void Deleteemployeeid()

            {
                //TODO

            }
            public static void Displayemployeeid()

            {
                //TODO
            }
        }
```

UML:

CLASS EMPLOYEES

+employee name : string
+employee id   : int
- employee salary: int
+employee desgination : string
 	
+ Addemployeeid() :void();
+ Editemployeeid() :void();
+ Deleteemployeeid ():void();
+ Displayemployeeid ():void();

AMAZON Class Customer:

Code:

```
class Customer
       {
           public string user name;
           private string password;
           public int mobile number;
           public stringemail;

           public static void Addcustomername()
           {
              //TODO
           }
            public static void Editcustomername()
           {
              //TODO
           }
           public static void Deletecustomername()

           {
              //TODO

           }
           public static void Displaycustomername()

           {
              //TODO
           }
       }
```

UML:

CLASS CUSTOMER

+customer user name :string
-customer password : string
+customer mobile number :string
- customer email id : string

+Addcusomername () : void();
+Editcustomername() : void();
+Deletecustomername(): void();
+Displaycustomername() :void();

AMAZON CLASS HOME:

Code:

```
class Home:
        {
           Public string your orders;
           private string your wishlist;
           public string deals;
           public  string help;

            public static void Addhomedeals()
           {
              //TODO
           }
            public static void Edithomedeals()
           {
              //TODO
           }
           public static void Deletehomedeals()

           {
              //TODO

           }
           public static void Displayhomedeals()

           {
              //TODO
           }
        }
```

UML:

CLASS HOME
+ home your orders :string - home your wishlist : string + home your deals :string +home your help :string

+ Addhomedeals(): Void();
+Edithomedeals() : void();
+Deletehomedeals() : void();
+Displayhomedeals() : void();

AMAZON CLASS Debit card:

Code:

```
class Debitcard
       {
          private int card ID;
          public string CardName;
          private int CardNumber;
          public int cardexp date;

public static void Adddebitcardid()
{
     //TODO
}
public static void Editdebitcardid()
{
     //TODO
}
public static void Deletedebitcardid()

{
     //TODO
 }
public static void Displaydebitcardid()

{
    //TODO
}
}
```

UML:

CLASS DEBIT CARD
-DebitCard ID : int + DebitCard CardName: string - DebitCard CardNumber : int +DebitCard cardexp date: int

+Add debitCardid () : void();
+Edit DebitCard id() : void();
+Delete DebitCard id() : void();
+ DisplaydebitCard id() : void();

Apollo Hospital:

CLASS Patient:

CODE:

```
class patient
       {
          public string Name;
          public string gender;
          public int age;
          public int mobile;

          public static void Addpatientname()
          {
             //TODO
          }
           public static void Editdpatientname()
          {
             //TODO
          }
          public static void Deletepatientname()

          {
             //TODO

          }
          public static void Displaypatientname()

          {
             //TODO
          }
       }
```

UML:

Class paient:

+Public Name : string
+Public gender :string
+Public age: int
+Public mobile : int

+Addpatientname() : void;
+Editpatientname() : void;
+Deletepatientname() : void;
+Displaypatientname() : void;

Class Hospital:

Code:

```
class hospital
        {
            public string Name;
            public string Address;
            public int mobile;
            public string services;

            public static void AddHospitalname()
            {
               //TODO
            }
             public static void Editdhospitalname()
            {
               //TODO
            }
             public static void Deletehospitalname()
            {
               //TODO
            }
             public static void Displaypatientname()

            {
               //TODO
            }
        }
```

UML:

Hospital
+Hospital name : string +Hospital address :string +Hospital mobile: int +public string: services

+Addhospitalname() : void();
+Edithospitalname() :   void();
+Deletehospitalname() : void();
+Displayhospitalname() :void()

Class Inpatient

Code:

```
class Inpatient
    {
        public string Name;
        public string wardname;
        public int roomid;
        public string status;

        public static void Addinpatientname()
        {
          //TODO
        }
         public static void Editdinpatientname()
        {
          //TODO
        }
        public static void Deleteinpatientname()

        {
          //TODO

        }
        public static void Displayinpatientname()

        {
          //TODO
        }
    }
```

UML:

CLASS INPATIEN
+Inpatient name: string +Inpatient wardname: string +Inpatientroomid :int +Inpatient: status :string

+Addinpatient name () :void();
+Editinpatientname () :void();
+Deleteinpatientname () :void();
+Displayinpatientname () :void();

Class Doctor:

Code:

```
class doctor
      {
          public string Name;
          public string specilisation;
          public int doctor id;
          public int doctor mobile;

           public static void Adddoctorname()
          {
             //TODO
          }
           public static void Editdoctoname ()
          {
             //TODO
          }
          public static void Deletedoctorname()

          {
             //TODO

          }
          public static void Displaydoctorname()

          {
             //TODO
          }

      }
```

UML:

Doctor
+doctor name : string +doctor specialization : string +doctor id : string +doctor mobile : int

+Adddoctorname(): void();
+Editdoctorname (): void();
+Deletedoctorname ():void;
+Displaydoctorname():void;

Class Medicine:

Code:

```
class Medicine
{
private string Name;
private int quantity;
private string expdate;
private string manufacturingdate;

public static void Addmedicinequanitity()
{
  //TODO
}
public static void Editdmedicinequantity()
{
  //TODO
}
public static void Deletemedicinequantity()

{
  //TODO
}
public static void Displaymedicinequantity()

{
  //TODO
}

}
```

UML:

Medicine

+medicine name : string
+quanity : int
+expdate :string
+manufacturingdate :string

+Addmedicinequantity() :void();
+Editmedicinequantity() : void();
+Deletemedicinequantity() : void();
+Displaymedicinequantity() : void();

Police Station:

Class Complaints:

code:

```
Class Complaints
{
        public string Complaint Name;
        public int complaint ID;
        public string complaint type;
        public string complaint description;

        public static void Addcomplaintname()
        {
          //TODO
        }
         public static void Editdcomplaintname()
        {
          //TODO
        }
         public static void Deletecomplaintname()

        {
          //TODO
        }
         public static void Displaycomplaintname()

        {
          //TODO
        }

        }
```

UML:

Complaint:

+comaplaint name : string
+comaplaint Id : int
+complainttype : string
+complaint description : string

+Addcomplaintname () : void();
+Editcomplaintname () : void();
+Delet complaintname () : void();
+Displaycomplaintname ():void ();

Class police:

Code:

```
 Class Complaints
 {
           public string police Name;
           public sring police ID;
           public string emailid;
           public int mobile;

public static void Addpoliceid()
           {
              //TODO
           }
 public static void Editpoliceid()
 {
 //TODO
 }
 public static void Deletepoliceid()

 {
 //TODO
 }
 public static void Displaypoliceid()

 {
 //TODO
 }

 }
```

UML:

Police:

+name : string
+id :string
+email :string
+mobile: int

+Addpoliceid (): void();
+Editpoliceid (): void();
+Deletepoliceid () : void();
+Displaypoliceid () : void();

Class crime:

Code:

```
class crime        {
          public int crimeid;
          public int criminal id;
          public string crime type;
          public string crime name;

          public static void Addcrimeid()
          {
             //TODO
          }
           public static void Editcrimeid()
          {
             //TODO
          }
          public static void Deletecrimeid()
          {
             //TODO
           }
          public static void Displaycrimeid()
         {
         //TODO
         }
     }
```

UML:

Crime class
+crime id : int +criminal id : int +crimetype :string +crimename :string

+Addcrimeid () : void();
+Editcrimeid () : void();
+Deletecrimeid () : void();
+Displaycrimeid () : void();

Class department:

Code:

```
Class departments           {
          public string department Name;
          public int departmentID;
          public string deparment place;
          public string department description;

           public static void Adddepartmentname()
          {
             //TODO
          }
           public static void Editddepartmentname()
          {
             //TODO
          }
           public static void Deletedepartmentname()

          {
             //TODO
          }
           public static void Displaydeparmentname()

          {
             //TODO
           }
}
```

UML:

Class department:
+department name : string +department id : int +department place : string +department description: string

+Adddepartment () : void();
+Editdepartment() : void();
+Deletedepartment() :void;
+Displaydepartment() :void;

Class case:

Code:

```
Class case

        Public string case Name;
        public int case ID;
        public case type ;
        public string case description;

        public static void Addcasename()
        {
          //TODO
        }
         public static void Editdcasename()
        {
          //TODO
        }
         public static void Deletecasename()

        {
          //TODO
        }
         public static void Displaycasename()

        {
          //TODO
        }

     }
```

UML:

Class Case
+case name : string +case id : int +case type : string +case description : string

+Addcase() : void();
+Editcase() : void();
+Deletecase() : void();
+Displaycase() : void();