

**ASSIGNMENT
BY
ANDE MANOHAR**

AMAZONE Class product

Code:

```
class Products
{
    private int price;
    private string brand;
    private string quantity;
    private 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();

AMAZONE Class employees

Code:

```
class Employee
{
    Private string emp name ;
    private string emp id;
    private int empsalry;
    private 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();

AMAZONE Class customer

Code:

```
class Customer
{
    private string user name;
    private string password;
    private int mobile number;
    private string email;
    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 : int
- customer email id : string

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

AMAZONE CLASS HOME

Code:

```
class Home
```

```

{
    private string your orders;
    private string your wishlist;
    private string deals;
    private 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	
<ul style="list-style-type: none"> - homeyour orders : string - home your wishlist : string - home your deals : string - home your help : string 	
<ul style="list-style-type: none"> + Addhomedeals(): Void(); + Edithomedeals() : void(); + Deletehomedeals() : void(); + Displayhomedeals() : void(); 	

AMAZONE CLASS Debit card

Code:

```

class Debitcard
{
    private int card ID;
    private string CardName;
    private int CardNumber;
    private int cardexp date;
    public static void Adddebitcardid()
    {

```

```

        //TODO
    }
    public static void Editdebitcardid()
    {
        //TODO
    }
    public static void Deleteddebitcardid()

    {
        //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();	

APPOLO HOSPITAL

CLASS Patient

CODE:

```

class patient
{
    private string Name;
    private string gender;
    private int age;
    private 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
-Patient Name : string - Patient gender :string - Patient age: int - Patient mobile : int
+Addpatientname() : void; +Editpatientname() : void; +Deletepatientname() : void; +Displaypatientname() : void;

Class Hospital
Code:
<pre> class hospital { private string Name; private string Address; private int mobile; public static void AddHospialname() { //TODO } public static void Editdhospitalname() { //TODO } public static void Displaypatientname() { //TODO } } </pre>

UML:	
Hospital	
-Hospital name : string - Hospitaladdress :string - Hospital mobile: int	
+Addhospitalname() : void(); +Edithospitalname() : void(); +Displayhospitalname() :void();	

Class Inpatient	
Code:	
<pre> class Inpatient { private string Name; private string wardname; private int roomid; private string status; public static void Addinpatientname() { //TODO } public static void Editdinpatientname() { //TODO } public static void Deleteinpatientname() { //TODO } public static void Displayinpatientname() { //TODO } } </pre>	
UML:	
CLASS INPATIEN	
-Inpatient name: string - Inpatient wardname: string - Inpatientroomid :int - Inpatient: status :string	
+Addinpatient name () :void();	

+Editinpatientname () :void(); +Displayinpatientname () :void(); +Displayinpatientname () :void();	
--	--

Class Doctor

Code:

```
class doctor
{
    private string Name;
    private string specilisation;
    private int doctor id;
    private 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(); +Editdoctoname (): 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 Addmedicinequantity()
    {
        //TODO
    }
    public static void Editmedicinequantity()
    {
        //TODO
    }
    public static void Deletemedicinequantity()
    {
        //TODO
    }
    public static void Displaymedicinequantity()
    {
        //TODO
    }
}

```

UML:

Medicine
-medicine name : string -quantity : int -expdate :string -manufacturingdate :string
+Addmedicinequantity() :void(); +Editmedicinequantity() : void(); +Deletemedicinequantity() : void(); +Displaymedicinequantity() : void();

POLICE SATION

Class Complaints

code:

```

Class Complaints {
    private string Complaint Name;

```

```

private int complaint ID;
private string complaint type;
private 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

-comaplain name : string - comaplain Id : int -complainttype : string - complaint description : string
+Addcomplaintname () : void(); +Editcomplaintname () : void(); +Delet complaintname () : void(); +Displaycomplaintname ():void ();

Class police

Code:

```

Class Complaints {
    private string police Name;
    private sring police ID;
    private string emailid;
    private 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 {
    private int crimeid;
    private int criminal id;
    private string crime type;
    private 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 -crimenam :string
+Addcrimeid () : void(); +Editcrimeid () : void(); +Deletecrimeid () : void(); +Displaycrimeid () : void();

Class department

Code:

```

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

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

    {
//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(); +Deletedeartment() :void; +Displaydepartment() :void;	

Class case	
Code:	
<pre> Class case { private string case Name; private int case ID; private case type ; private string case description; public static void Addcasename() { //TODO } public static void Editdcasename() { //TODO } public static void Deletecasename() { //TODO } public static void Displaycasename() { //TODO } } </pre>	
UML:	
Class Case	
-case name : string -case id : int -case type : string	

-case description : string	
+Addcase() : void(); +Editcase() : void(); +Deletecase() : void(); +Displaycase() : void();	