

Day – 5 Morning Assignment
By
Vamsi Krishna Mandapati

Amazon

Products

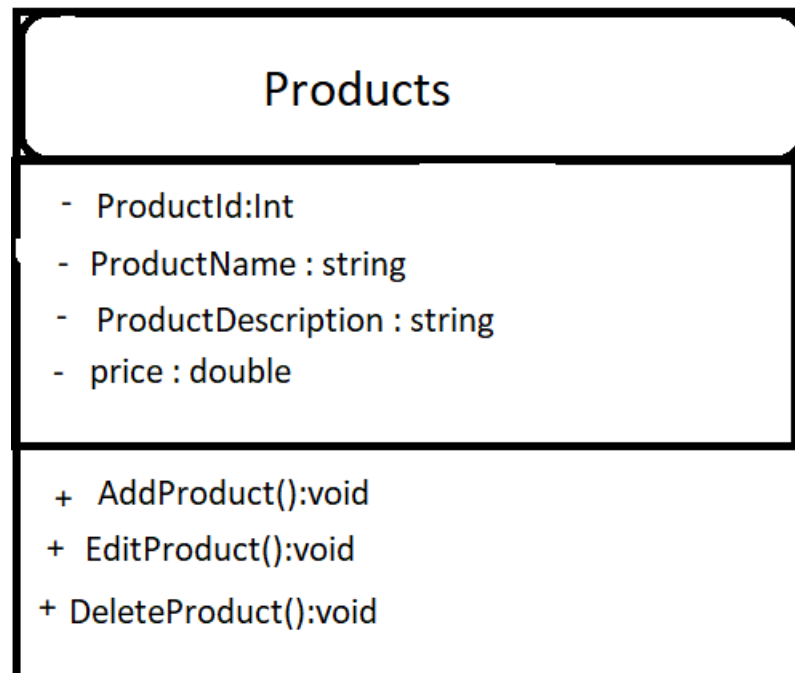
Code

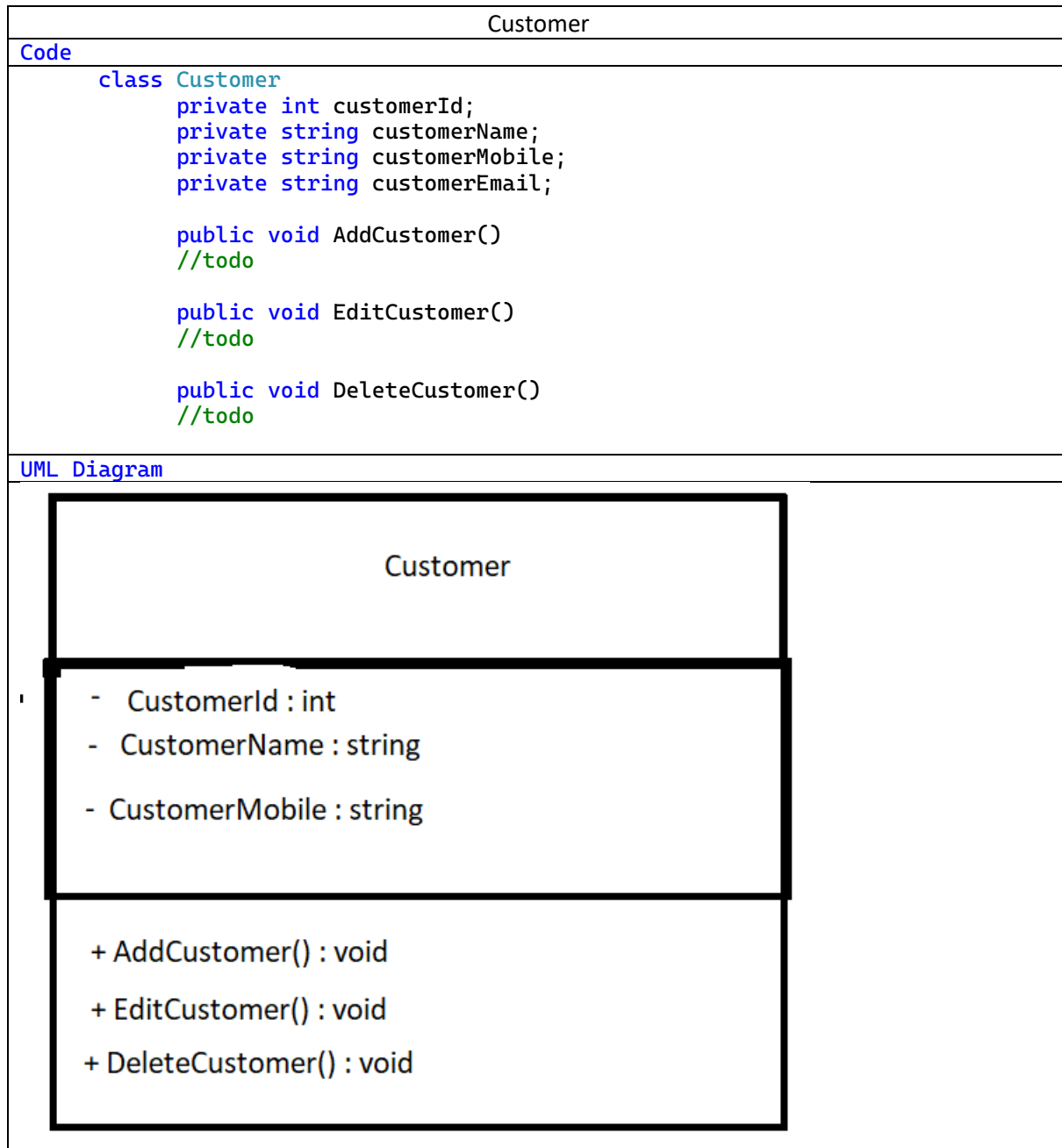
```
class Products
{
    private int productId;
    private string productName;
    private string productDescription;
    private double price;

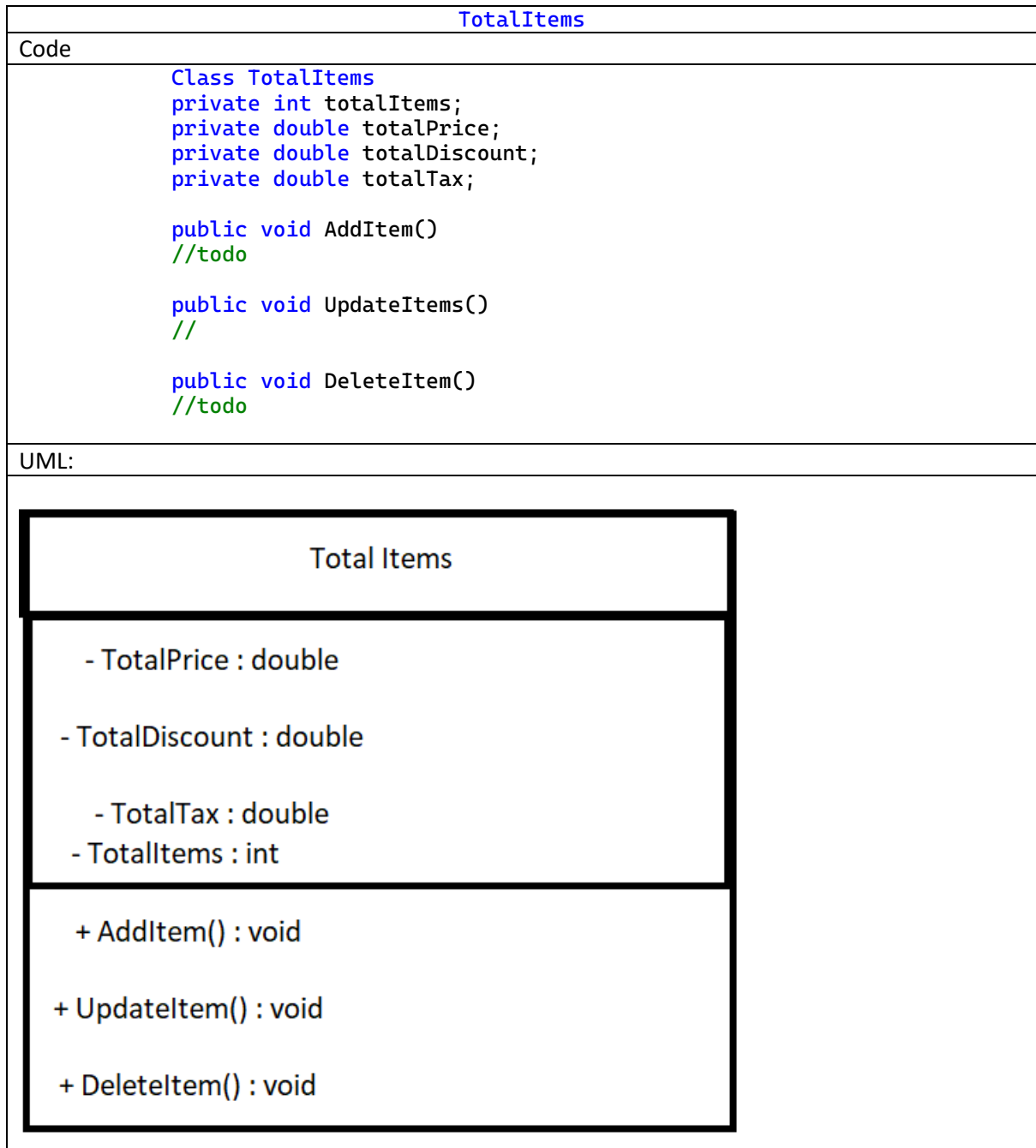
    public void AddProduct()
    //todo
    public void EditProduct()
    //todo

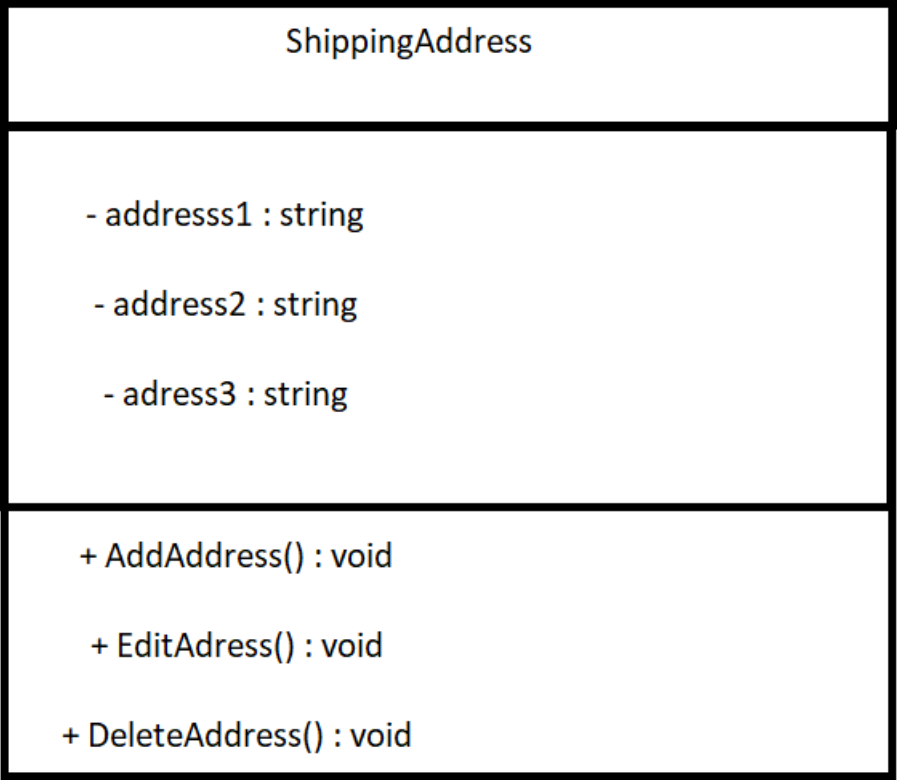
    public void DeleteProduct()
    //todo
}
```

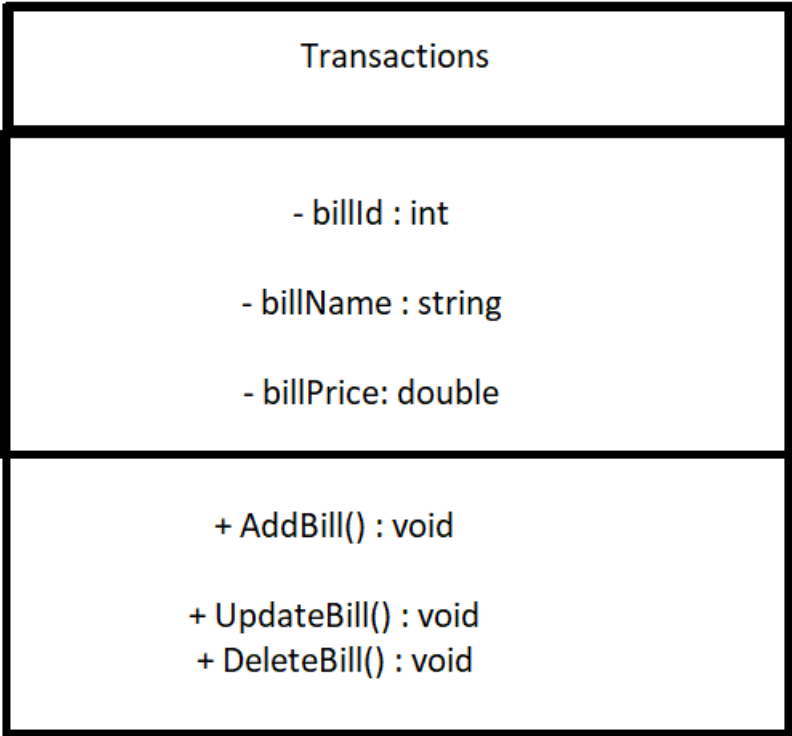
UML Diagram



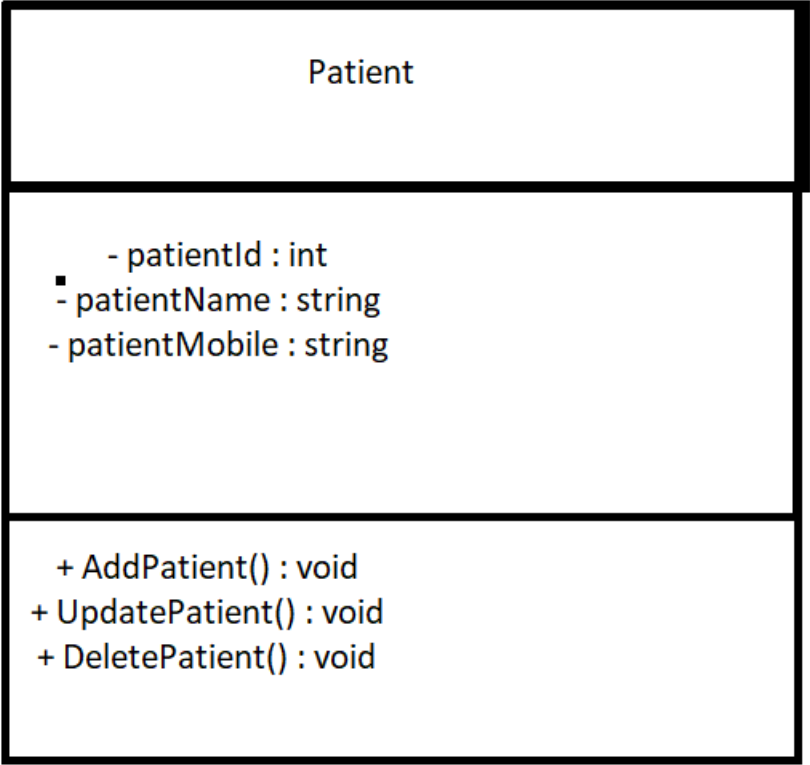




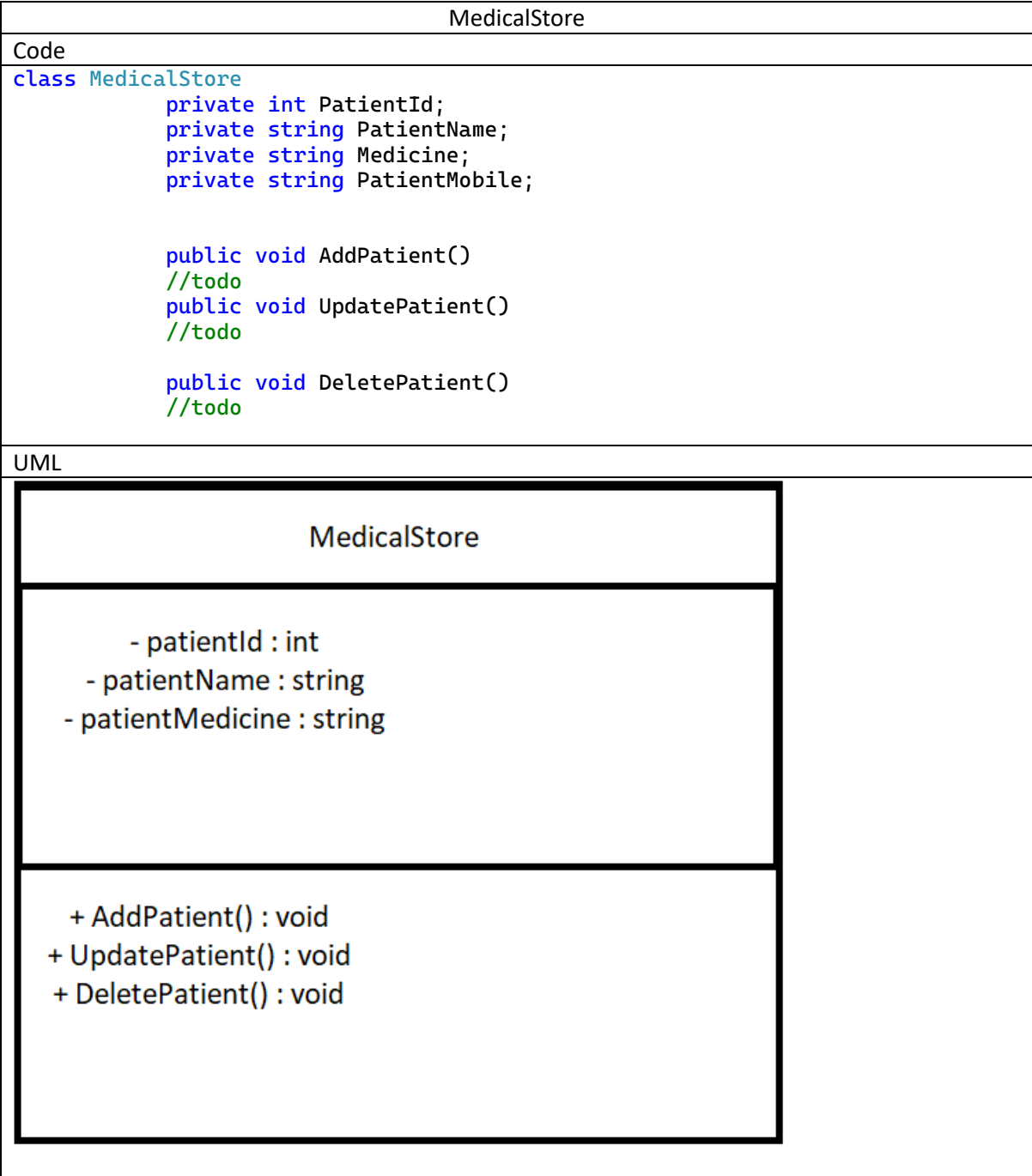
| ShippingAddress |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Code |
| <pre> class ShippingAddress private string address1; private string address2; private string address3; public void AddAddress() //todo public void UpdateAddress() //todo public void DeleteAddress() //todo </pre> |
| UML |
|  <pre> classDiagram class ShippingAddress { -addresss1 : string -address2 : string -adress3 : string +AddAddress() : void +EditAdress() : void +DeleteAddress() : void } </pre> <p>The UML class diagram for ShippingAddress is structured as follows:</p> <ul style="list-style-type: none"> Class Name: ShippingAddress Attributes (Private): <ul style="list-style-type: none"> - addresss1 : string - address2 : string - adress3 : string Operations (Public): <ul style="list-style-type: none"> + AddAddress() : void + EditAdress() : void + DeleteAddress() : void |

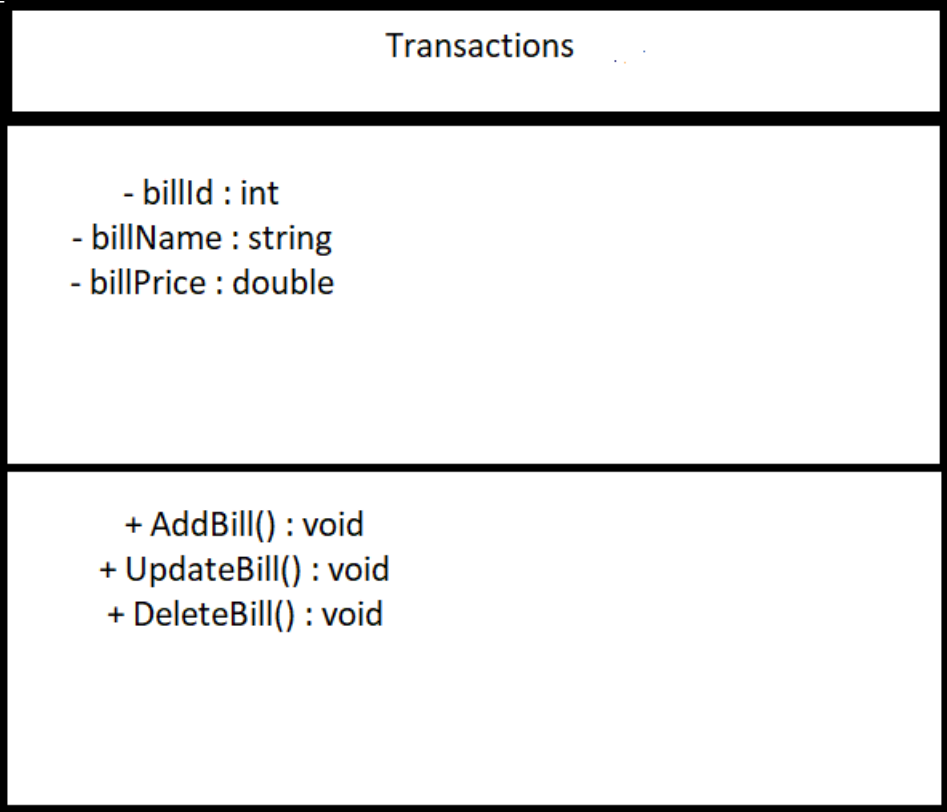
| Transactions | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Code | |
| <pre>class Transactions private int billId; private string billName; private double billPrice; private double billDiscount; private double billTax; public void AddBill() //todo public void UpdateBill() //todo public void DeleteBill() //todo</pre> | |
| UML | |
|  <pre>classDiagram class Transactions { -billId : int -billName : string -billPrice : double +AddBill() : void +UpdateBill() : void +DeleteBill() : void }</pre> <p>The UML class diagram for the Transactions class is structured as follows:</p> <ul style="list-style-type: none">Class Name: TransactionsAttributes (Private):<ul style="list-style-type: none">- billId : int- billName : string- billPrice: doubleOperations (Public):<ul style="list-style-type: none">+ AddBill() : void+ UpdateBill() : void+ DeleteBill() : void | |

| Doctors |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Code |
| <pre>class Doctors { private int DoctorId; private string DoctorName; private string DoctorDesignation; public void AddDoctorName() //todo public void EditDoctorName() //todo public void DeleteDoctorName() //todo }</pre> |
| UML |
| <div><div>Doctors</div><div><div>- doctorId : int</div><div>- doctorName : string</div><div>- doctorDesignation</div></div><div><div>+ AddDoctor() : void</div><div>+ EditDoctor() : void</div><div>+ DeleteDoctor() : void</div></div></div> |

| Patient |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Code |
| <pre>class Patient private int PatientId; private string PatientName; private string PatientMobile; private string PatientEmail; public void AddPatient() //todo public void UpdatePatient() //todo public void DeletePatient() //todo</pre> |
| UML |
|  <pre>classDiagram class Patient { -patientId : int -patientName : string -patientMobile : string +AddPatient() : void +UpdatePatient() : void +DeletePatient() : void }</pre> <p>The UML class diagram for the Patient class is structured as follows:</p> <ul style="list-style-type: none">Class Name: PatientAttributes:<ul style="list-style-type: none">- patientId : int- patientName : string- patientMobile : stringOperations:<ul style="list-style-type: none">+ AddPatient() : void+ UpdatePatient() : void+ DeletePatient() : void |

| BillCounter | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Code | |
| <pre>class BillCounter private int ItemNo; private double totalPrice; private double totalDiscount; private double totalTax; public void AddItem() //todo public void UpdateItems() // public void DeleteItem() //todo</pre> | |
| UML | |
| <div><div>BillCounter</div><div><div>- itemNo : int</div><div>- totalPrice : double</div><div>- totalDiscount : double</div></div><div><div>+ AddItem() : void</div><div>+ Updateltem() : void</div><div>+ Deleteltem() : void</div></div></div> | |

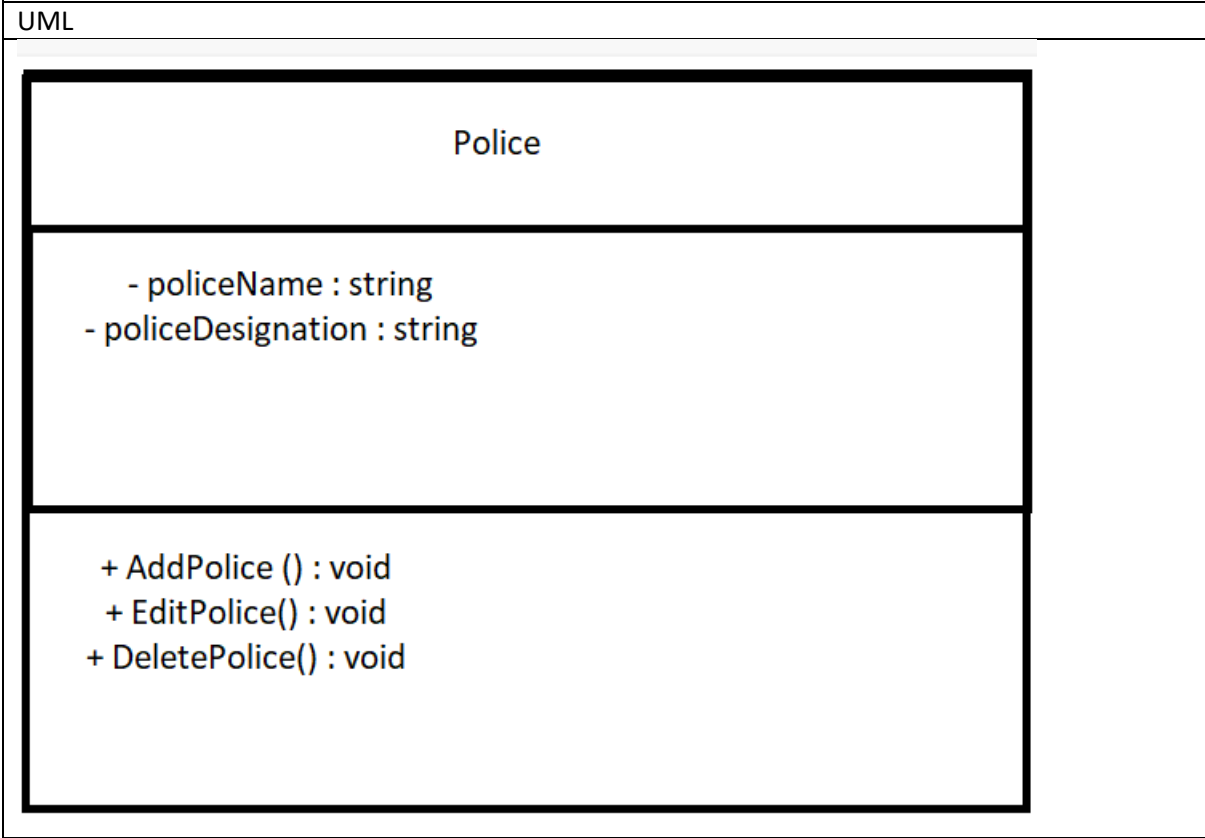


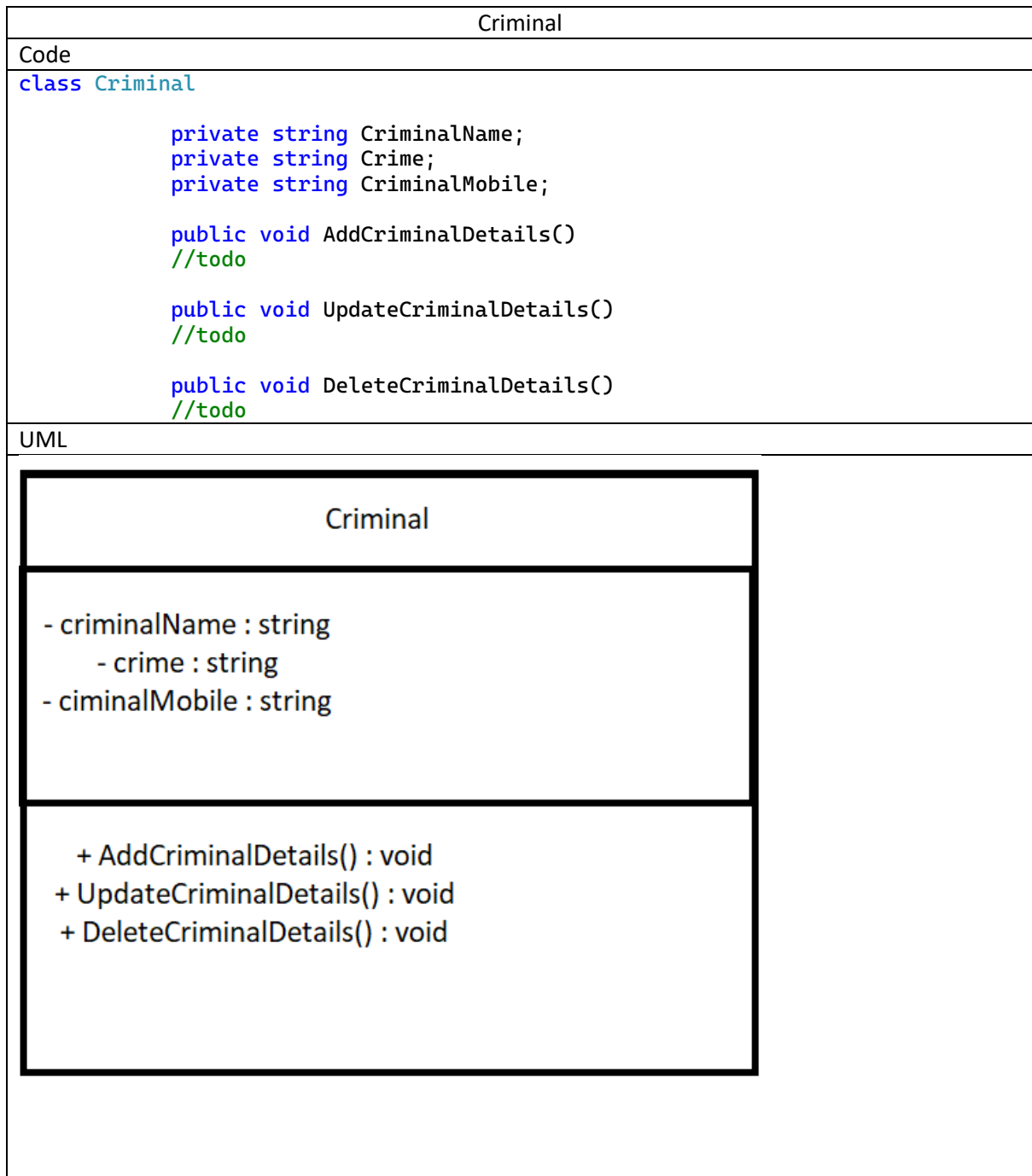
| Transactions | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Code | |
| <pre>class Transactions private int billId; private string billName; private double billPrice; private double billDiscount; private double billTax; public void AddBill() //todo public void UpdateBill() //todo public void DeleteBill() //todo</pre> | |
| UML | |
|  <pre>classDiagram class Transactions { -billId : int -billName : string -billPrice : double +AddBill() : void +UpdateBill() : void +DeleteBill() : void }</pre> <p>The UML class diagram for the Transactions class is shown. It features a class box with the name 'Transactions' at the top. The box is divided into three compartments. The top compartment is empty. The middle compartment contains the private attributes: '- billId : int', '- billName : string', and '- billPrice : double'. The bottom compartment contains the public methods: '+ AddBill() : void', '+ UpdateBill() : void', and '+ DeleteBill() : void'.</p> | |

| |
|---------------|
| PoliceStation |
|---------------|

| |
|--------|
| Police |
|--------|

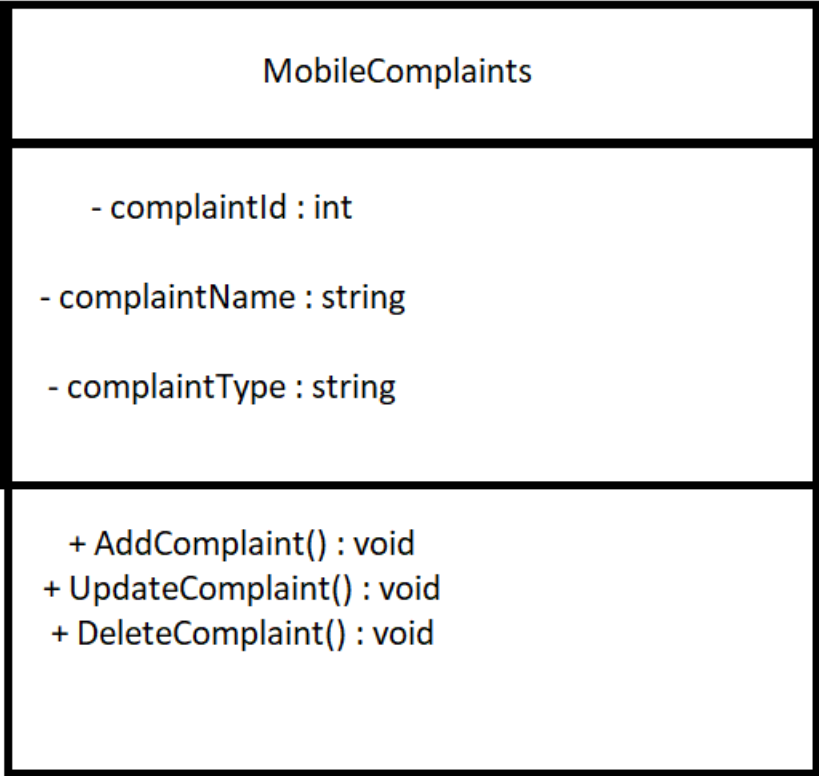
| |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Code |
| <pre>class Police private string PoliceName; private string PoliceDesignation; public void AddPoliceName() //todo public void EditPoliceName() //todo public void DeletePoliceName() //todo</pre> |





| CrimeRecords |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Code |
| <pre>class CrimeRecords private int RecordNo; private int totalRecords; public void AddRecord() //todo public void UpdateRecord() // public void DeleteRecord() //todo</pre> |
| UML |
|  <pre>classDiagram class CrimeRecords { -recordNo : int -totalRecords : int +AddRecord() : void +UpdateRecord() : void +DeleteRecord() : void }</pre> <p>The UML class diagram for CrimeRecords is a rectangular box divided into three horizontal compartments. The top compartment is labeled 'CrimeRecords'. The middle compartment contains two private attributes: '- recordNo : int' and '- totalRecords : int'. The bottom compartment contains three public methods: '+ AddRecord() : void', '+ UpdateRecord() : void', and '+ DeleteRecord() : void'.</p> |

| TrafficRules | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Code | |
| <pre>class TrafficRules private int RuleId; private string RuleHeading; public void AddRule() //todo public void UpdateRule() //todo public void DeleteRule() //todo</pre> | |
| UML | |
| <pre>classDiagram class TrafficRules { -ruleId : int -ruleHeading : string +AddRule() : void +UpdateRule() : void +DeleteRule() : void }</pre> | |

| MobileComplaints |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Code |
| <pre>class MobileComplaints private int ComplaintId; private string ComplainterName; private string ComplaintType; public void AddComplaint() //todo public void UpdateComplaint() //todo public void DeleteComplaint() //todo</pre> |
| UML |
|  <pre>classDiagram class MobileComplaints { -complaintId : int -complaintName : string -complaintType : string +AddComplaint() : void +UpdateComplaint() : void +DeleteComplaint() : void }</pre> <p>The UML class diagram for MobileComplaints shows a class with three private attributes: complaintId (int), complaintName (string), and complaintType (string). It also has three public methods: AddComplaint() (void), UpdateComplaint() (void), and DeleteComplaint() (void).</p> |