|  |
| --- |
| **Day 5 (28-01-2022) Assignment by M.Pallavi** |

**Amazon**

|  |
| --- |
| Product class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class Program  {  class product  {  private int pid;  private string pname;  private float pprice;  private string invoice;  private string brand;  public string ProductDetails()  {  }  public void ProductAlternative()  {  }  public string LinkDetails()  {  }  public void Productmaterial()  {  }  public void ProductVendor()  {  }  }  } |
| **UML Diagram:**   |  | | --- | | Product | | -name : string  -pprice : float  -invoice :string  -brand :string | | +ProductDetails() : String  +productAlternatives(): Void  +Productmaterial() : Void  +LinkDetails() : string  +productVendor() : String | |

|  |
| --- |
| Customer class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class Program  {  class Customer  { private int cid;  private string cfirstname;  private string clastname;  private string cPhonenumber;  private string cAddress;  public void CustomerDetails()  {  }  public void PlaceOrder()  {  }  public void CancelOrder()  {  }  public void Billpayment()  {  }  public void Returnorder()  {  }  }  } |
| **UML Diagram:**   |  | | --- | | Customer | | -cid : string  -cfirstname : string  -clastname :string  -phonenumber :int  -cAddress :String | | +customerDetails() : String  +PlaceOrder() : Void  +CancelOrder() : Void  +Billpayment() : Void  +Returnorder() : String | |

|  |
| --- |
| Employee class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class Program  {  class Employee  { private int eid;  private string ename;  private float esalary;  private int ephonenumber;  private string edesignation;  public void updateproductDetails()  {  }  public void developcode()  {  }  public void ensurepayments()  {  }  public void UpdateCustomerDetails()  {  }  public void updateOrder()  {  }  }  } |
| **UML Diagram:**   |  | | --- | | Employee | | -eid : int  -ename : string  -salary :float  -ephoneno :int  -edesignation :String | | + updateproductDetails(): String  + developcode() : Void  + ensurepayments() : Void  + UpdateCustomerDetails(): string  + updateOrder () : String | |

|  |
| --- |
| Shipment class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class Program  {  class shipmentDetails  { private int Sid;  private string Sdate;  private string stype;  private string scost;  private string shipmentdays;  public void invoicedetails()  {  }  public void shipmentpayment()  {  }  public void productdetails()  {  }  public void customerdetails()  {  }  public void address()  {  }  }  } |
| **UML Diagram:**   |  | | --- | | shipmentDetails | | -sid : int  -sdate : date  -scost :float  -shipmentdays :int  -stype :String | | + InvoiceDetails() : String  +Shipmentpayment () : Void  + Productdetails() : string  + CustomerDetails() : string  + Address() : string | |

|  |
| --- |
| OrderDetails class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class program  {  class OrderDetails  { private int oid;  private string odate;  private string productname;  private string customername;  private date deliverydate;  public void ExchangeOrder()  {  }  public void ReturnOrder()  {  }  public void Cancelorder()  {  }  public void OrderDelivery()  {  }  public void OrderFeedback()  {  }  }  } |
| **UML Diagram**:   |  | | --- | | Order class | | - oid : int  - odate : date  - productname :string  - customername :string  - deliverydate : date | | + ExchangeOrder (): String  + ReturnOrder() : Void  + Cancelorder() :Void  + OrderDelivery() : int  +OrderFeedback() : String | |

**Hospital**

|  |
| --- |
| Doctor class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class program  {  class Doctor  {  private int id;  private string name;  private string specialization;  private int Dept;  private date salary;  public void PrescribeMed()  {  }  public void ChecksReport()  {  }  public void DrawSalary()  {  }  public void PrescribeTest()  {  }  public void OrderFeedback()  {  }  }  } |
| UML Diagram:   |  | | --- | | Doctor class | | - id : int  - name : sttring  - specialization :string  - Dept :string  - salary :float | | +ExchangeOrder() : String  +ReturnOrder() : Void  +Cancelorder() :Void  +OrderDelivery() : int  +OrderFeedback() : String | |
| Patient class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class program  {  class patient  {  private int ptid;  private string ptname;  private string ptage;  private string paddress;  private date pgender;    public void pdetails()  {  }  public void paycharges()  {  }  public void patientadmit()  {  }  public void patientdischarge()  {  }  public void pmedicine()  {  }  } } |
| UML Diagram:   |  | | --- | | Patient class | | - ptid : int  - ptname : strting  - ptage :int  - paddress :string  - pgender :String | | + pdetails() : string  + paycharges() : int  + patientadmit() :string  + patientdischarge(): int  + pmedicine() : string | |

|  |
| --- |
| Admin class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class Admin  {  private int staffid;  private string staffname;  private string staffage;  private string staffphoneno;  private date staffgender;    public void registerpatient()  {  }  public void dischargespatient()  {  }  public void updatepatientinfo()  {  }  public void roomsallocate()  {  }  public void doctorappointment()  {  }  }  }  } |
| UML Diagram:   |  | | --- | | Admin class | | - staffid : int  - staffname : string  - staffage :int  - staffphone :int  - staffgender :String | | +registerpatient () :string  + dischargespatient() : int  + updatepatientinfo() :string  + roomsallocate() : int  + doctorappointment() : string | |

|  |
| --- |
| Test class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class Test  {  private int testno;  private string testtype;  private string patientname;  private string patientno;  private int bill;  public void addorder()  {  }  public void updatearesult()  {  }  public void addtestcharges()  {  }  public void performtest()  {  }  public void returnreport()  {  }  }  }  } |
| UML Diagram:   |  | | --- | | Test class | | - testno : int  - testtype : float  - patientname : int  - patientno : int  - bill : string | | + addorder() :Void  + updatearesult() : int  + addtestcharges():string  + performtest() : int  + returnreport() : string | |

|  |
| --- |
| Pharmacy class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class pharmacy  {  private int medicineNo;  private string medicineName;  private date Expirydate;  private float amount;  private int billno;  public void supplymedicine()  {  }  public void billpayment()  {  }  public void checkmedicineAva()  {  }  public void generatetestbill()  {  }  public void updatemedicine()  {  }  }  }  } |
| UML Diagram: |
| |  | | --- | | Pharmacy class | | - medicineno : int  - medicinename : string  - expirydate : date  - amount : float  - billno : int | | +supplymedicine() :Void  +billpayment() :int  +checkmedicineAva() :string  +generatetestbill() :int  +updatemedicine() : string | |

**Police Station**

|  |
| --- |
| police class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class pharmacy  {  private int PNo;  private string PName;  private string P\_age;  private string pService;  private int Pdesignation;  public void respondstoCall()  {  }  public void enrollscase()  {  }  public void connecttoCourt()  {  }  public void punishprisoners()  {  }  public void checksprisoners()  {  }  }  }  } |
| UML Diagram:   |  | | --- | | Police class | | - PNo : int  - PName : string  - P\_age : int  - pService : string  - Pdesignation : String | | + respondstoCall() :Void  + enrollscase() :int  + connecttoCourt() :string  + punishprisoners() :int  + checksprisoners() : string | |

|  |
| --- |
| Prisoner class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class prisioner  {  private int PNo;  private string PName;  private string P\_age;  private string Pcaseno;  private int Paddress;    public void PrisionerPunishment()  {  }  public void AttendsCourt()  {  }  public void LogintoRegister()  {  }  public void ConsultLawyer()  {  }  public void Meetsfamilymem()  {  }  }  } |
| UML Diagram:   |  | | --- | | Prisoner class | | - PNo : int  - PName : string  - P\_age : int  - Pcaseno : int  - Paddress : String | | + PrisionerPunishment():Void  + AttendsCourt() :int  + Logintoregister() :string  +Consutlawyer() :void  + Meetsfamilymem() :string | |

|  |
| --- |
| Clerk class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class clerk  {  private int id;  private string c\_Name;  private string c\_age;  private string C\_phonrno;  private int c\_address;    public void enrollspoliceinfo()  {  }  public void enrollsprisionerinfo()  {  }  public void loginregister()  {  }  public void logoutregister()  {  }  public void attendcalls()  {  }  }  }}  UML Diagram:   |  | | --- | | Order class | | - id : int  - c\_name : string  - c\_age : int  - c\_phoneno : int  - c\_address : String | | +enrollspoliceinfo() :Void  +enrollsprisionerinfo():int  +loginregister() :string  +loginregister() : int  + attendcalls() : string | |

|  |
| --- |
| User class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class user  {  private int U\_no;  private string U\_name;  private string U\_phoneno;  private int U\_address;  private int U\_age;    public void complaint()  {  }  public void viewCasehistory()  {  }  public void giveFIRs()  {  }  public void attendscourt()  {  }  public void searchprisioner()  {  }  }  } |
| UML Diagram:   |  | | --- | | Order class | | - U\_no : int  - U\_name : string  - U\_phoneno : int  - U\_address : int  - U\_age : String | | + Complaint() :Void  + ViewCasehistory() :int  + GiveFIRs() :string  + Attendscourt() : int  + SearchPrisioner() : string | |

|  |
| --- |
| Witness class |
| Code |
| using System;  namespace ConsoleApp1  {  internal class  {  class Witness  {  private int w\_no;  private string w\_name;  private string w\_phoneno;  private int w\_address;  private int w\_age;    public void complaints()  {  }  public void updateinfo()  {  }  public void makeFIRs()  {  }  public void attendscourt()  {  }  public void caseinfo()  {  }  }  }  } |
| **UML Diagram:**   |  | | --- | | witness class | | - w\_no : int  - w\_name : string  - w\_phoneno : int  - w\_address : int  - w\_age : String | | + complaints() :Void  + updateinfo() :string  + makeFIRs() :string  + attendscourt() :int  + caseinfo() :string | |