AIR UNIVERSITY

Assignment#4

Subject : Object Oriented Programming

Instructor : Prof Dr Abdul Hameed

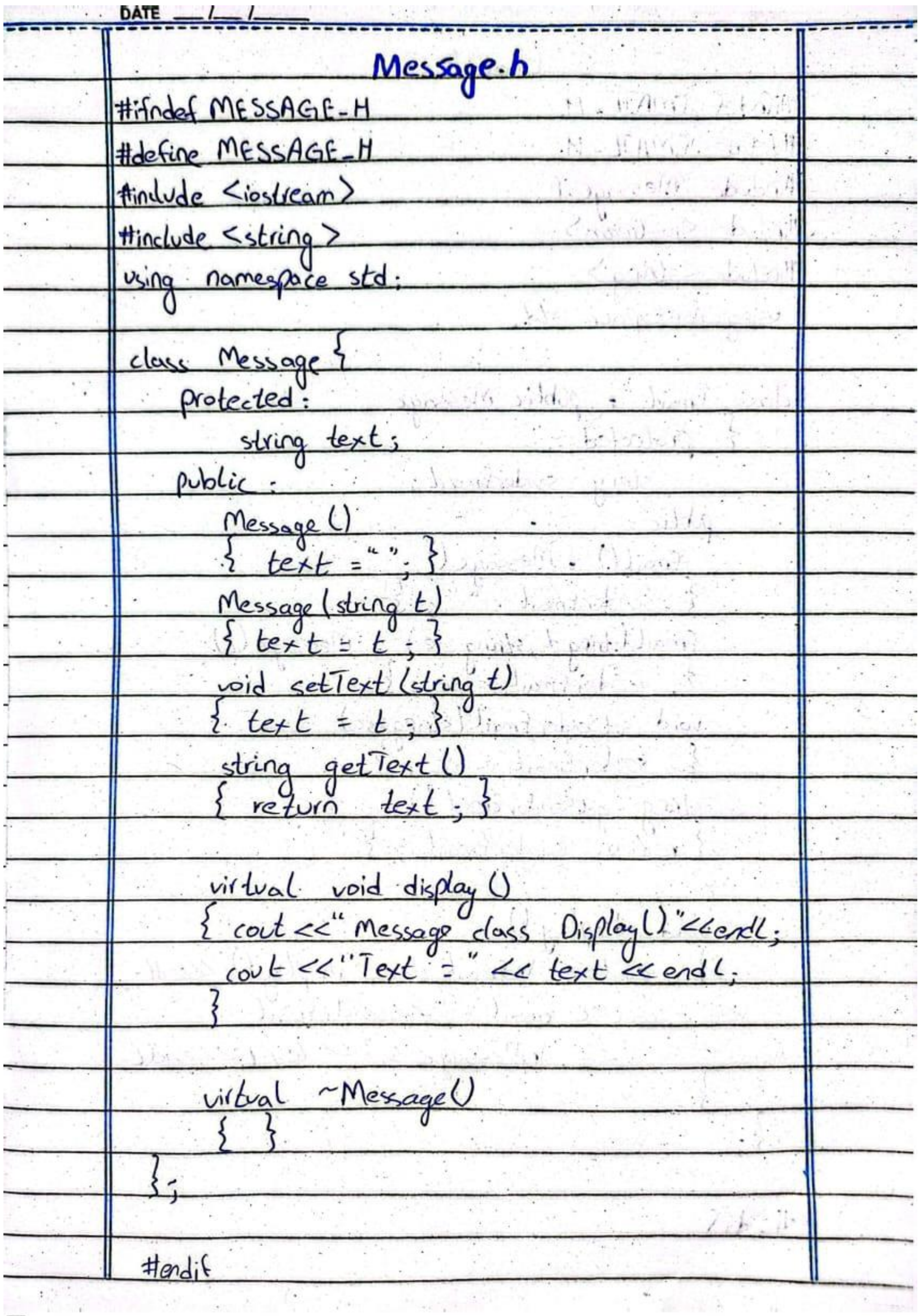
Department of Computing and Artificial Intelligence

Name: MUHAMMAD TALHA | Roll No. 241103

BSIT-F-24-A

Date of Submission : 15/06/2025

Q#1



DATE ___/___/_____

Message.h

```
#ifndef MESSAGE-H
#define MESSAGE_H
#include <iostream>
#include <string>
using namespace std;

class Message {
    protected:
        string text;
    public:
        Message()
        { text = " "; }
        Message(string t)
        { text = t; }
        void setText(string t)
        { text = t; }
        string getText()
        { return text; }

        virtual void display()
        { cout <<" Message class Display()"<<endl;
          cout <<"Text = " << text << endl;
        }

        virtual ~Message()
         { }
  };

#endif
```

DATE: __ /__ /____

Email.h

```
#ifndef EMAIL_H
#define EMAIL_H
#include "Message.h"
#include <iostream>
#include <string>
 using namespace std;

class Email : public Message
    { protected :
          string  senderEmail;
      public :
        Email() : Message()
        {  senderEmail = "";}
        Email(string t, string se) : Message(t)
        {  senderEmail = se;}
        void setSenderEmail(string se)
        {  senderEmail = se ;}
        string getSenderEmail ()
        { return senderEmail ; }

      void display ()
      {   cout<<"\nEmail class Display()"<<endl;
          cout<<"Email = "<<senderEmail
               << "\nMessage = "<< text <<endl;
      }
   };

#endif
```

DATE: __/__/____

SecureEmail.h

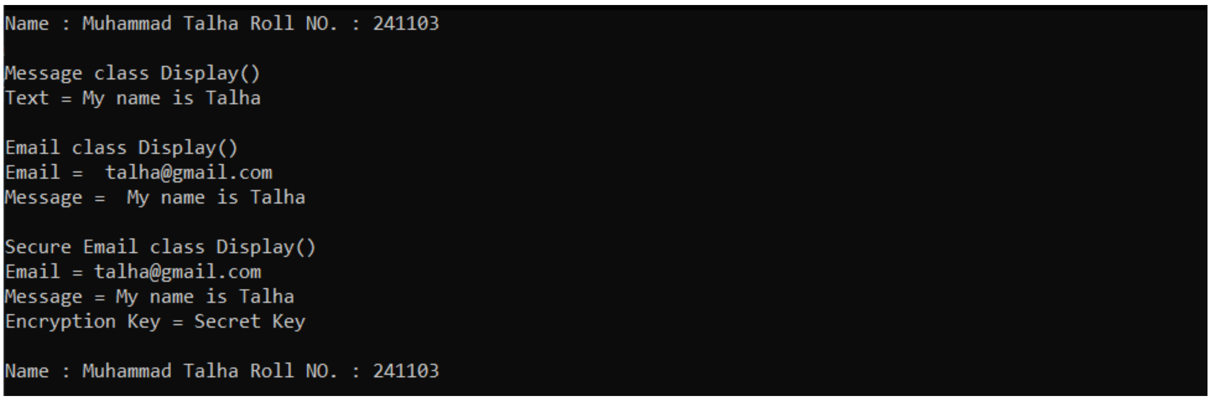
```
#ifndef SECUREEMAIL_H
#define SECUREEMAIL_H
#include "Email.h"
#include <iostream>
#include < string>
using namespace std;

class SecureEmail : public Email {
    private :
        string encryptionKey;
    public ::
      SecureEmail() : Email()
      { encryptionKey = ""; }
     SecureEmail (string t, string se, string ek)
                          : Email (t, se)
     { encryptionKey = ek ; }
     void setEncryptionKey (string ek)
     { encryptionKey = ek ; }
     string getEncryptionKey ()
     { return encryptionKey; }

     void display ()
     { cout<<"\n Secure Email class Display()"<<endl;
      cout << "Email = "<< senderEmail << "\nMessage = "
          << text << "\n Encryption key = "
          << encryptionKey <<endl ;
     }
  };
#endif
```

DATE __/__/__

Main.cpp

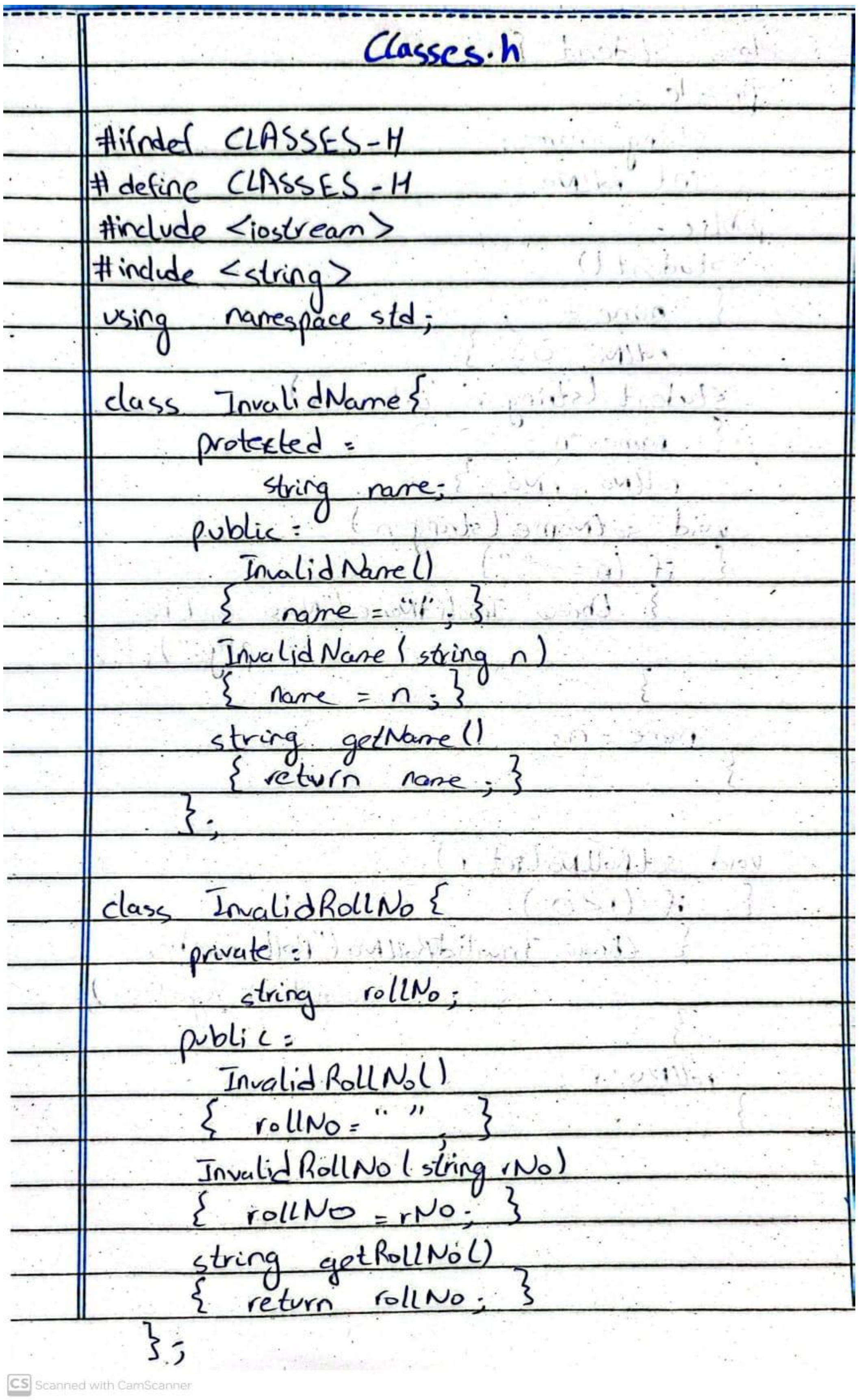
```
#include <iostream>
#include <string >
#include "Message.h"
#include "Email.h"
#include "SecureEmail.h"
using namespace std;

int main()
  {     cout<<"Name : Muhammad Talha Roll No. 24103"<<endl;

    Message* message1 = new Message ("My name is Talha");
  Message* message2=new Email ("My name is Talha",
                          "talha@gmail.com");

  Message* message3 = new SecureEmail ("My name is Talha",
                          "talha@gmail.com",
                            "Secret Key" );
    message1 -> display ();
    message2 ->display ();
    message3 -> display ();

    delete message1;
    delete message2;
    delete message3;

    cout<<"\n Name :Muhammad Talha
             Roll No. : 24103 \n"<<endl;
    return 0;
}
```

OUTPUT



```
Name : Muhammad Talha Roll NO. : 241103

Message class Display()
Text = My name is Talha

Email class Display()
Email =  talha@gmail.com
Message =  My name is Talha

Secure Email class Display()
Email = talha@gmail.com
Message = My name is Talha
Encryption Key = Secret Key

Name : Muhammad Talha Roll NO. : 241103
```

Q#2


Classes.h

```
#ifndef CLASSES-H
# define CLASSES-H
#include <iostream>
#include <string>
using namespace std;

class InvalidName{
        protected :
            string name;
      public :
          InvalidName()
        {   name = "";}
         InvalidName ( string n)
         {  name = n ;}
         string getName()
          { return name ; }
      };

class InvalidRollNo {
       private :
           string rollNo ;
      public :
         InvalidRollNo()
        {   rollNo = " " ;  }
        InvalidRollNo ( string rNo)
        {  rollNo = rNo;  }
        string getRollNo()
        {  return rollNo ;  }
   };
```

DATE: __ /__ /____

```
class Student {
 private :
     string name;
     int rollNo;
 public :
   Student()
   { name = "";
     rollNo = 0; }
   Student (string n, int rNo)
   { name= n;
     rollNo = rNo; }
   void setName (string n)
   { if (n== "")
      { throw InvalidName ("Name cannot be
                                  empty.");
      }
     name = n;
   }

 void setRollNo(int r)
 {   if (r<0)
     { throw InvalidRollNo ("Roll number
                          cannot be negative.");
     }
   rollNO=r;
 }
```

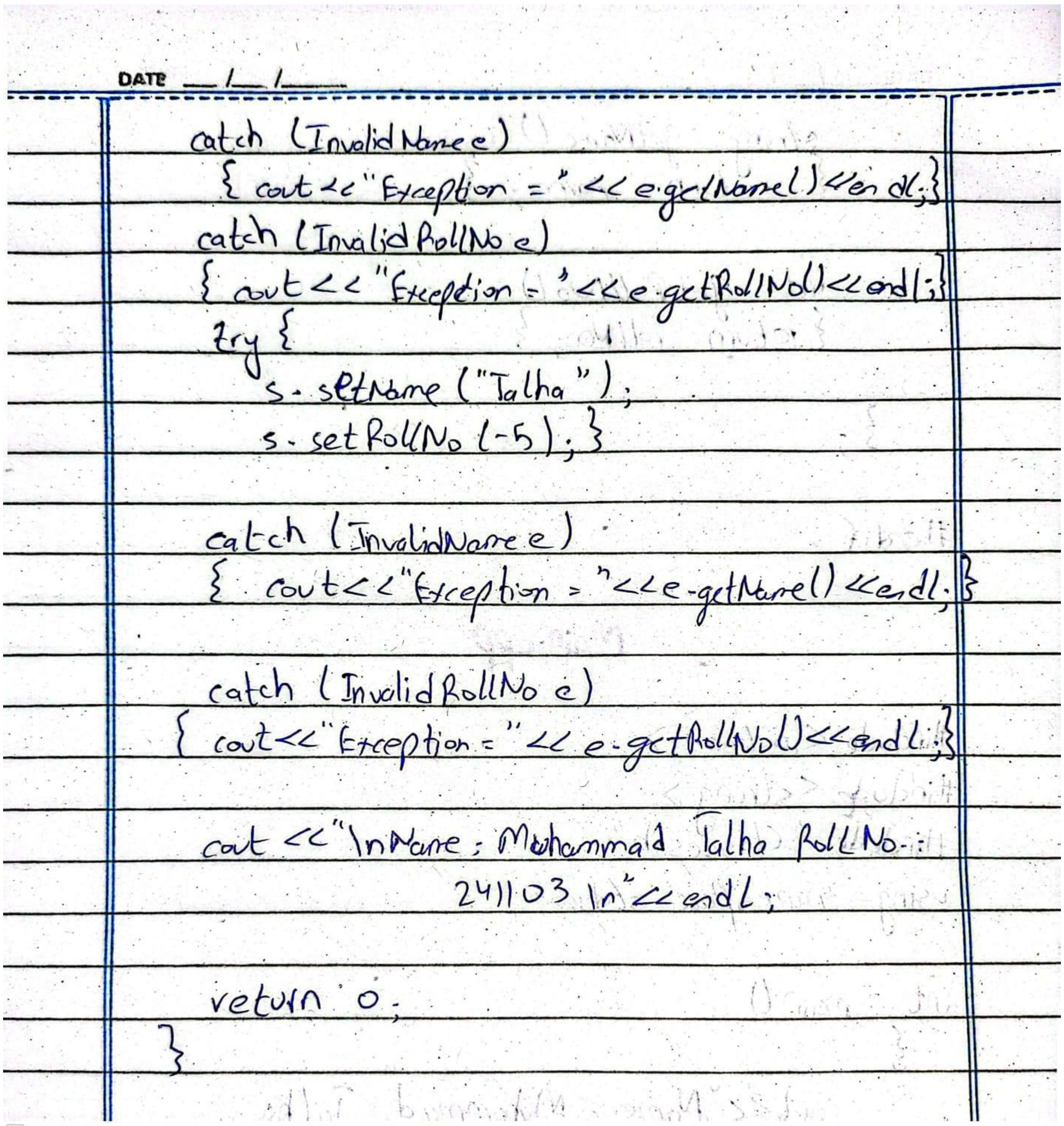
DATE: ___/___/_____

```
        string getName()
        { return name; }

        int getRollNo()
        { return rollNo; }

    };

#endif
```

Main.cpp

```
#include <iostream>
#include <string>
#include "Classes.h"
using namespace std;

int main()
  {
        cout<<"Name : Muhammad Talha
              Roll No. : 241103 \n" << endl;

        Student s;

       try {
            s.setName ("");
            s.setRollNo (10); }
```

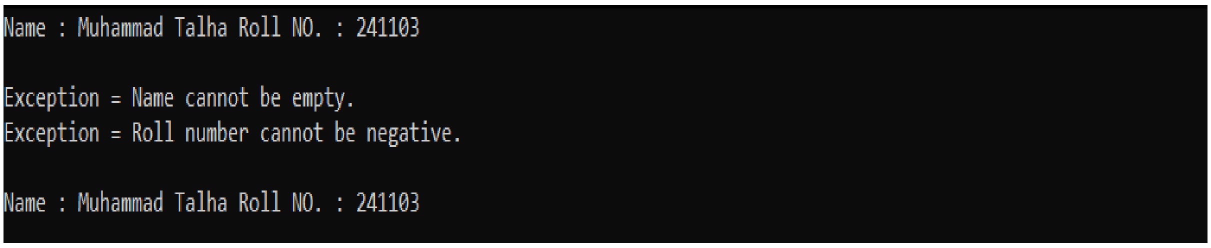
Scanned with CamScanner



DATE ___/___/______

```
catch (InvalidName e)
 { cout << "Exception = " << e.getName() << endl;}
catch (InvalidRollNo e)
{ cout << "Exception = " << e.getRollNo() << endl;}
try {
    s.setName ("Talha");
    s.setRollNo (-5); }

 catch (InvalidName e)
  { cout << "Exception = " << e.getName() << endl; }

 catch (InvalidRollNo e)
{ cout << "Exception = " << e.getRollNo() << endl;}

 cout << "\nName : Muhammad Talha Roll No. :
                  241103 \n" << endl;

 return 0;
}
```

OUTPUT



```
Name : Muhammad Talha Roll NO. : 241103

Exception = Name cannot be empty.
Exception = Roll number cannot be negative.

Name : Muhammad Talha Roll NO. : 241103
```

Q#3

Classes.h

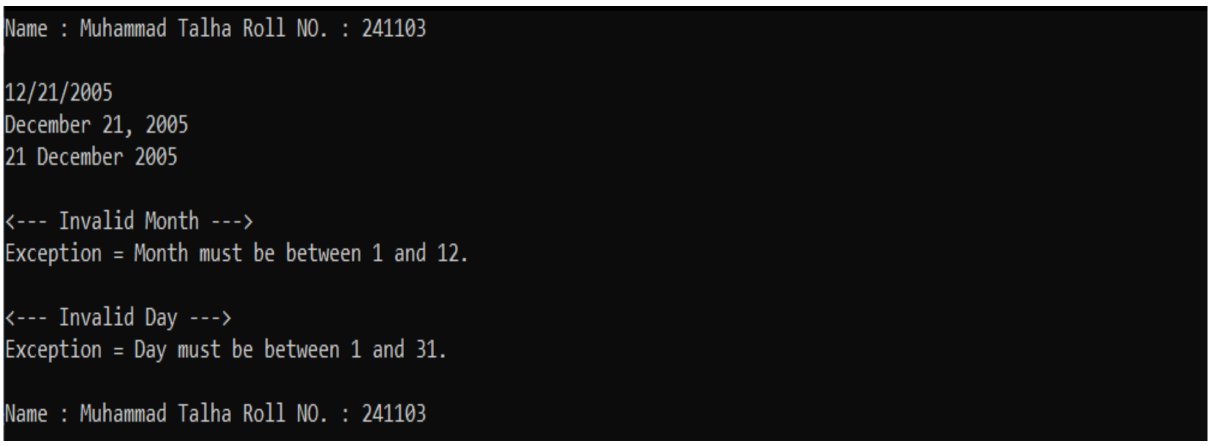
```
#ifndef CLASSES_H
#define CLASSES_H
#include <iostream>
#include <string>
using namespace std;

class InvalidDay {
private:
   string day;
public:
   InvalidDay()
   {
      day = "";
   }
   InvalidDay(string d)
   {
      day=d;
   }
   string getDay()
   {
      return day;
   }
};

class InvalidMonth {
private:
   string month;
public:
   InvalidMonth()
   {
      month = "";
   }
   InvalidMonth(string m)
   {
      month=m;
   }
```

```
    string getMonth()
    {
        return month;
    }
};

class Date {
private:
    int day, month, year;
    string monthNames[13] = { "","January", "February", "March", "April",
"May", "June",
                        "July", "August", "September", "October", "November",
"December" };


public:
    Date()
    {
        year = 0;
    }
    Date(int d, int m, int y)
    {
        setDay(d);
        setMonth(m);
        year = y;
    }

    void setDay(int d)
    {
        if (d < 1 || d > 31)
        {
            throw InvalidDay("Day must be between 1 and 31.");
        }
        day = d;
    }

    void setMonth(int m)
    {
        if (m < 1 || m > 12)
        {
            throw InvalidMonth("Month must be between 1 and 12.");
        }
        month = m;
    }

    void setYear(int y)
    {
        year = y;
    }

    void printFormat1()
    {
        cout << month << "/" << day << "/" << year << endl;
    }

    void printFormat2()
    {
        cout << monthNames[month] << " " << day << ", " << year << endl;
    }

    void printFormat3() {
        cout << day << " " << monthNames[month] << " " << year << endl;
    }
};

#endif
```

Main.cpp

```
#include<iostream>
#include<string>
#include"Classes.h"
using namespace std;

int main() {
    cout << "Name : Muhammad Talha Roll NO. : 241103\n" << endl;
    try {
        Date date1(21, 12, 2005);
        date1.printFormat1();
        date1.printFormat2();
        date1.printFormat3();

        cout << "\n<--- Invalid Month --->"<<endl;

        Date date2(10, 13, 2025);



    }
    catch (InvalidDay e)
    {
        cout << "Exception = " << e.getDay() << endl;
    }
    catch (InvalidMonth e)
    {
        cout << "Exception = " << e.getMonth() << endl;
    }

    try {
        cout << "\n<--- Invalid Day --->" << endl;

        Date date3(35, 6, 2025);

    }
    catch (InvalidDay e)
    {
        cout << "Exception = " << e.getDay() << endl;
    }
    catch (InvalidMonth e)
    {
        cout << "Exception = " << e.getMonth() << endl;
    }

    cout << "\nName : Muhammad Talha Roll NO. : 241103\n" << endl;
    return 0;
}
```

OUTPUT



```
Name : Muhammad Talha Roll NO. : 241103

12/21/2005
December 21, 2005
21 December 2005

<--- Invalid Month --->
Exception = Month must be between 1 and 12.

<--- Invalid Day --->
Exception = Day must be between 1 and 31.

Name : Muhammad Talha Roll NO. : 241103
```

Q#4

Exceptions.h

```
#ifndef EXCEPTIONS_H
#define EXCEPTIONS_H
using namespace std;

class InvalidProductDataException {
public:
   void showError()
   {
     cout << "Error: Invalid or missing product data!" << endl;
   }
};

class ProductNotFoundException {
public:
   void showError()
   {
     cout << "\nError: Product not found in inventory!" << endl;
   }
};

#endif
```

Product.h

```
#ifndef PRODUCT_H
#define PRODUCT_H

#include <iostream>
#include <string>
#include "Exceptions.h"
using namespace std;


class Product {
protected:
   string name;
   double price;
```

```
    int quantity;
public:
    Product()
    {
        name = "";
        price = 0;
        quantity = 0;
    }
    Product(string n, double p, int q)
    {
        if (n=="" || p < 0 || q < 0)
        {
            throw InvalidProductDataException();
        }
        name = n;
        price = p;
        quantity = q;
    }

    virtual void displayInfo() = 0;
    virtual double calculateDiscountedPrice() = 0;

    string getName()
    {
        return name;
    }
    double getPrice()
    {
        return price;
    }
    int getQuantity()
    {
        return quantity;
    }

    virtual ~Product() {}
};

#endif
```

ElectronicsProduct.h

```
#ifndef ELECTRONICSPRODUCT_H
#define ELECTRONICSPRODUCT_H
#include "Product.h"
using namespace std;

class ElectronicsProduct : public Product {
   int warrantyYears;
public:
   ElectronicsProduct()
   {
      warrantyYears = 0;
   }
   ElectronicsProduct(string n, double p, int q, int w) : Product(n, p, q)
   {
      warrantyYears = w;
   }
   void setWarrantyYears(int w)
   {
      warrantyYears = w;
   }
   int getWarrantyYears()
   {
      return warrantyYears;
   }
   void displayInfo()
   {
     cout << "\nElectronics Product = "<<endl;
     cout << "Name = " << name << "\nPrice = " << price << "\nQuantity = " <<
quantity << "\nWarranty = " << warrantyYears << " years"<<endl;
   }

   double calculateDiscountedPrice()
   {
      return price * 0.9;
   }
};

#endif
```

GroceryProduct.h

```
#ifndef GROCERYPRODUCT_H
#define GROCERYPRODUCT_H
#include "Product.h"
using namespace std;

class GroceryProduct : public Product {
  string expiryDate;
public:
   GroceryProduct()
   {
      expiryDate = "";
   }
   GroceryProduct(std::string n, double p, int q, string e) : Product(n, p, q)
   {
      expiryDate = e;
   }
   void setExpiryDate(string e)
   {
      expiryDate = e;
   }
   string getExpiryDate()
   {
      return expiryDate;
   }

   void displayInfo()
   {
     cout << "\nGrocery Product:\n";
     cout << "Name = " << name << "\nPrice = " << price<< "\nQuantity = " <<
quantity<< "\nExpiry Date = " << expiryDate << endl;
   }

   double calculateDiscountedPrice()
   {
      return price * 0.95;
   }
};
```

```
#endif
```

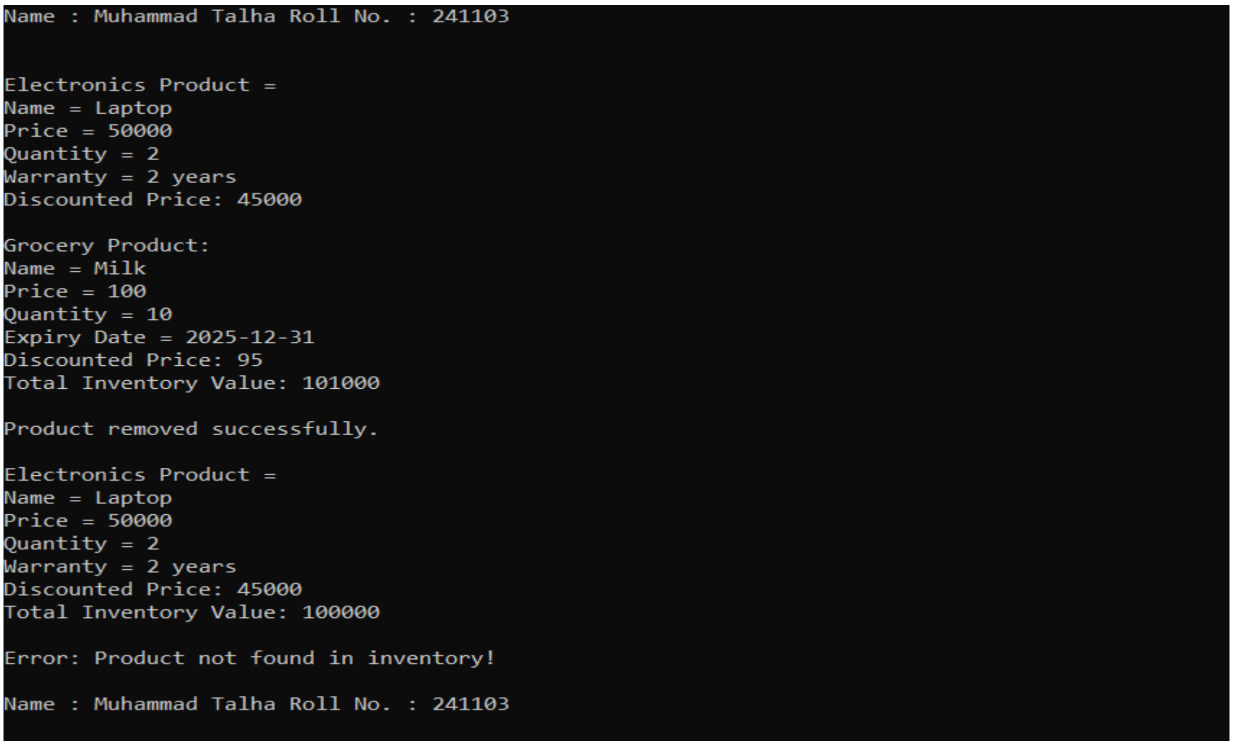
Inventory.h

```
#ifndef INVENTORY_H
#define INVENTORY_H

#include "Product.h"
using namespace std;

template <class T>
class Inventory {
   Product* products[10];
   int count;
public:
   Inventory()
   {
      count = 0;
   }
   Inventory(int c)
   {
      count = c;
   }

   void addProduct(Product* p)
   {
      if (count < 10)
      {
         products[count++] = p;
      }
      else
      {
         cout << "Inventory is full.\n";
      }
   }

   void removeProduct(string pname)
   {
      int index = -1;
      for (int i = 0; i < count; i++)
      {
            if (products[i]->getName() == pname)
            {
                index = i;
                break;
            }
        }
        if (index == -1)
        {
            throw ProductNotFoundException();
        }
        else
        {
            delete products[index];
            for (int i = index; i < count - 1; i++)
            {
                products[i] = products[i + 1];
            }
            count--;
            cout << "\nProduct removed successfully."<<endl;
        }
    }

    void displayInventory()
    {
        if (count == 0)
        {
           cout << "Inventory is empty.\n";
        }
        for (int i = 0; i < count; i++)
        {
            products[i]->displayInfo();
            cout << "Discounted Price: " << products[i]->calculateDiscountedPrice()
<< endl;
        }
    }

    double calculateTotalValue()
    {
        double total = 0;
        for (int i = 0; i < count; i++)
        {
            total += products[i]->getPrice() * products[i]->getQuantity();
        }
        return total;
    }

    ~Inventory()
    {
        for (int i = 0; i < count; i++)
        {
            delete products[i];
        }
    }
};

#endif
```

Main.cpp

```
#include <iostream>
#include "Inventory.h"
#include "ElectronicsProduct.h"
#include "GroceryProduct.h"
#include "Exceptions.h"
using namespace std;

int main() {
    cout << "Name : Muhammad Talha Roll No. : 241103\n" << endl;

    Inventory<Product> store;

    try {
        Product* p1 = new ElectronicsProduct("Laptop", 50000, 2, 2);
        Product* p2 = new GroceryProduct("Milk", 100, 10, "2025-12-31");

        store.addProduct(p1);
        store.addProduct(p2);

        store.displayInventory();
        cout << "Total Inventory Value: " << store.calculateTotalValue() << endl;

        store.removeProduct("Milk");
        store.displayInventory();
        cout << "Total Inventory Value: " << store.calculateTotalValue() << endl;

        store.removeProduct("NonExisting");

    }
    catch (InvalidProductDataException& e)
    {
        e.showError();
    }
    catch (ProductNotFoundException& e)
    {
        e.showError();
    }

    cout << "\nName : Muhammad Talha Roll No. : 241103\n" << endl;

    return 0;
}
```

OUTPUT



```
Name : Muhammad Talha Roll No. : 241103


Electronics Product =
Name = Laptop
Price = 50000
Quantity = 2
Warranty = 2 years
Discounted Price: 45000

Grocery Product:
Name = Milk
Price = 100
Quantity = 10
Expiry Date = 2025-12-31
Discounted Price: 95
Total Inventory Value: 101000

Product removed successfully.

Electronics Product =
Name = Laptop
Price = 50000
Quantity = 2
Warranty = 2 years
Discounted Price: 45000
Total Inventory Value: 100000

Error: Product not found in inventory!

Name : Muhammad Talha Roll No. : 241103
```