

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

	Message.h	
	#Hodef MESSAGIE-H	
	#define MESSAGE_H NOTAWAY	
	Hindude <iosteam></iosteam>	
	Hinclude <string></string>	
	using namespace std:	10-11
	class Message &	
	protected: someth sides - hours	
	string text;	
	public : hand also produce	
	Message ()	
	is text = ; Support in the state of	
	Message (string t)	
٠.		
	void setText (string t)	
A	{ text = t ; } in the bar	
	string getText() { return text; }	
	{ return text; }	
	virtual void display ()	
	{ cout << " Message class Display()" << cont.;	
	cout << "Text = " << text << endl;	
	3 Year 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
	Almost and the second of the second	
	virtual ~Message()	
	{ }	
	7.15.11.	
	#andif	

Email h	
Historia Historia Historia Historia	
Hodefine EMAIL-H	
#include "Message.h"	
Hindude Gostream>	
Hinclude <string></string>	
using namespace std:	
t specially into	
dass Emil : public Message	ľ
of protected:	
string sender Email;	
Ablic: (). specific	
Email() = Message(). { sender Email = ""; }	
{ sender Email = "(s) } note) process	
fmail (string t string se) = Message (t)	
{ sender frail = set; }	
void set Sender Email (string se)	1
{ sender Email = se ; }	1
string getsender Email ()	
{ return sender frail; }.	
yinder (- voit diplay ().	
void display ()	
} cont<< "In Fmail class Display ()" Kendl-	
contec " Email = " << sender Email	
<< "InMessage = " Let t 2 Lendl.	100
$\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}_{\mathcal{L}}}}}}}}}}$	
} -	
Hend if	

10.510	Score Email h	
	#Ander SECUREFMATL-H	
7/10	# define SECUREEMAIL - H	Fall
	#include "Email.h"	
	#include <iostreem></iostreem>	
	#include < string>	
	using narrespace std;	
		*.
	class Secure Fmail = public Fmail }	1
	private :	
	string encryption Key:	2.
	Laboration of the second	
	Soure Email() : Email()	
	{ encryptionkey = ""; }	
	Secure Fmail (string & string se string ek)	
3	: Email (t se)	
1	{ encryption Key = ek . }	
	upid set Encryption Key (string ek)	
	{ exyptionKey = ek ;}	727
	string getEncryption Key ()	
	{ return encryptionkey; }	
	0, 0,	
	void display U	
	{ cout < 1 /2 Secure Smail class Display V'Men	V:
	+ 4"fm=il = "16 codocfmil <="\nMessage=	3,
-	< < ancryption Yey = " < < ancryption Yey < < ancryption Yey = "	
7	<< manufacture scool	
5	3	
1	3:	
-	1 16	
	Hency	11

Main-cpp	
#include <iostream></iostream>	
#include <string> History 10072 smile!</string>	
#include "Message.h"	
Hindude "Fmail. h"	
Hirclude "Secule frail. h"	1
using namespace std;	
int main()	
{ cost << "Name: Muhammad Talha Roll No. 74/103 21enc	11
the southern south	1
Message " message 1 = new Message ("My nape is Talha")	,
Message " message 1 = new Message ("My nape is Talha") Message "message 2= new Frail ("My name is Talha")	
"talke@gmil.com");	
Commission Deposit Line 1 , 100	170.0
Message " message 3: = new Secure Email ("My more is Talha	,
talho@gmail.com",	0.
"Secret Key");	
message 1 - display ();	
ressage 2 > display ();	
message 3-7 display()	
delete mssage1;	
delete message 2	
delete messagi3:	
cate "In Nane: Muhammad Talka"	
Roll No. : 24/103 \n" < Candl -	
veturn 0;	
1	

```
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
```

	Classes.h
	Alloder CLASSES-H
	# define CLASSES-H
	Hirclude <iostream></iostream>
	#include <string></string>
	using namespace std;
	O CONTO
	class InvalidName { in the state
	protexted:
	String name;
٠	public: (a probe) a mon : bien
	Inalid Namel)
	11 } name = 01/3 31 000 }
	Invalid Nane (string n)
	{ Name = n ; }
	string getName()
	{ return name; }
-	1 5 1110 1111 5 1110 1111 1111
-	class Inalid Roll No { (05)
	private (2)) and solveit west and
	string rollNo;
	public:
7 1	Invalid Roll Nol)
N	{ rollNo = " " }
	Invalid Roll No 1 string (No)
-	{ rollNo = rNo. }
4	string getRollNoL)
	{ return roll No; }

	class Student (
	Private:	
	string name: Harris Istalia	
	int pullo; H-2128/10 yest 1	
	police consider student	
	Student () Student	1
2	mme = "" bleaming out piece	
	rollNo=0.}	1
	Student (string n int Mo)	
	$ \begin{cases} \text{Name} = N; \end{cases} $	
	rollNo=1No; }	
	void setName (string n)	4
	{ if (n = = "")	
	Ethrow Invalid Name l'Name aumot be	2
	(paid) multienpty.");	
	} \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
	name = Ω ; 1) and Ω	
	and and a	
	void sethollyolint ()	1000
	{ if (r<0) Allower lines and	
	{ throw Invalid Roll No l'Roll number	
	cannot be regative ")	
-	ξ	
	rollNo=r; Walled Follows	
-	}	-
#		#

DATE:_	
	string getName 0
	E return name; 3
	Carlo Steparit Refer
1.160	inti getrourou
	{return roll No; }
	Les Sterne ("Talla")
3	= 1. (d.) on sign = = .
#endi	(Company of Noto
16:00	3 . Outer 2" (1 mtm " "222 c. a Haire () -
	Main-cpp
	Constitution (Second Selection) does
Hindud	e Cia-beam?
	e <string></string>
#includ	Le "Classesih" to " word > 1
The same of the	parespace std; 0000
using	Turespace sta;
ı ot-	~.0/\
ints	main ()
	coit<< "Name: Muhammad Talha
	All All Sunsal "
	Roll No.: 24/103 \n" << endl;
	- 1 1 1
	Students:
	17 {
	5-setName (""); 5-setfoll No (10);}
	4 Cotholl No (10) - 3

	DATE//	
	catch (Involid Namee) { cout < " Exception = " < c e.gr(Namel) < en of;} catch (Invalid PollNo e) { out < < "Exception = " < x e.gr(RollNoll< and);}	
	catch (Invalid PollNoe)	
	{ out < 2 "Freption - " < x e get Roll Noll < Land []	
	s-set RollNo ("Talha"); s-set RollNo (-5);}	
	s-set RollNo (-5);}	
	catch (InvalidNamee) { cout <pre>cout</pre> <pre>cout</pre> <pre>"</pre> <pre>ce-getName() Kendl;</pre>	
* 1 * 1	{ cout<1"(treption = "<1e-getNamel) Kendl.	3
	catch (Involid RollNo e) { cout<< "Exception = " <pre>"</pre> cont <pre>cont</pre> <pre>cont</pre> <pre>"</pre> <pre>cont</pre> <pre>cont</pre> <pre>"</pre> <pre>cont</pre> <pre>cont<td></td></pre>	
	cout< "Exception = "Le egethollNoll <andliss< td=""><td></td></andliss<>	
3 - 1445	## 보다 그리고 하는 그는 사람들이 하는 사람들이 사람들이 하실하게 되어 하는 것은 사람들이 되어 되어 보다면 생각하는 것이다. ## 그렇게 되었다면 ###	1
	cost <<"In Nane: Muhammala Talha Poll No.:1	
	241103 In Zeendl;	
	veturo;	
	}	
	Les More Manned Wille	

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;</pre>
  }
  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();</pre>
```

```
date1.printFormat3();
  cout << "\n<--- Invalid Month --->"<<endl;</pre>
  Date date2(10, 13, 2025);
catch (InvalidDay e)
  cout << "Exception = " << e.getDay() << endl;</pre>
catch (InvalidMonth e)
  cout << "Exception = " << e.getMonth() << endl;</pre>
try {
  cout << "\n<--- Invalid Day --->" << endl;
  Date date3(35, 6, 2025);
catch (InvalidDay e)
  cout << "Exception = " << e.getDay() << endl;</pre>
catch (InvalidMonth e)
  cout << "Exception = " << e.getMonth() << endl;</pre>
cout << "\nName : Muhammad Talha Roll NO. : 241103\n" << endl;</pre>
return 0;
```

}

```
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</pre>
```

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;</pre>
   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";</pre>
   cout << "Name = " << name << "\nPrice = " << price<< "\nQuantity = " <<
quantity<< "\nExpiry Date = " << expiryDate << endl;
  }
  double calculateDiscountedPrice()
    return price * 0.95;
};
```

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";</pre>
    }
  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;</pre>
    }
  }
  void displayInventory()
    if (count == 0)
      cout << "Inventory is empty.\n";</pre>
    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;</pre>
    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;
}</pre>
```

```
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
```