National University of Computer and Emerging Sciences



Lab Exercise 07

For

Object Oriented Programming Lab

|  |  |
| --- | --- |
| Course Instructor(s) | Mr. Usman Ghous |
| Lab Instructor(s) | Mr. Usman Ghous |
| Semester | Spring 2021 |

**FAST School of Computing**

# Instructions:

1. Make a word document with the naming convention “SECTION\_ LAB#\_ROLLNO” and put all your source code and snapshots of its output in it. Make sure your word file is formatted properly.
2. Plagiarism is strictly prohibited.
3. Do not discuss solutions with one another.

# Useful links

|  |
| --- |
| **Question#1** |

**(Polynomial Class)** Develop class Polynomial. The internal representation of a Polynomial is an

array of terms. Each term contains a coefficient and an exponent, e.g., the term 2x4 has the

coefficient 2 and the exponent 4. Develop a complete class containing proper constructor and

destructor functions as well as set and get functions The class should also provide the following

overloaded operator capabilities:

a) Overload the addition operator (+) to add two Polynomials.

b) Overload the subtraction operator (-) to subtract two Polynomials.

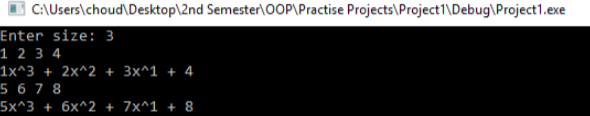
c) Overload the assignment operator to assign one Polynomial to another.

d) Overload the multiplication operator (\*) to multiply two Polynomials.

e) Overload the addition assignment operator (+=), subtraction assignment operator (-=), and

multiplication assignment operator (\*=).

Sample Input:



**Question#2 Operator Overloading**

Define a class **Matrix** to represent rows × cols matrix. r (row) and c (column) will be passed as

parameters to constructor of class Matrix and dynamic memory will be allocated to double

pointer according r and c.

1. Overload operators for addition (use "+" operator for addition) and subtraction (use "-"

operator for subtraction) of two matrices.

2. Overload operators for pre-increment (use "++" operator. This operator will increment

(all elements of matrix by) 1.

3. Overload operators for post-increment (use "++" operator. This operator will increment

(all elements of matrix by) 1.

4. Overload operators for pre-decrement (use "--" operator. This operator will decrement (all

elements of matrix by) 1.

5. Overload operators for post-decrement (use "--" operator. This operator will decrement

(all elements of matrix by) 1.

6. Overload insertion ">>" to input all elements of matrix.

7. Overload extraction "<<" operator to output all elements on console.

8. Overload less than operator "<" for two matrices. This operator will return true if the sum

of all elements of matrix is less than second. i.e. A < B.

9. Overload less than operator ">=" for two matrices. This operator will return true if all

elements of matrix A is greater than or equal to second. i.e. A >= B. If any one element is

smaller than B at same location, it will return false.

10. Overload unary operators “\*” that will return the product of all elements of a matrix.

**Note: Size of matrix A will be same as size of B for binary operators**.

Write a driver program to test your class.

**Question#3 Operator Overloading**

Write a class **Factorial** to overload ! operator to find factorial of an integer object. Write a driver program to test your class. Show input and output results on console.

**Question#4 Operator Overloading**

Suppose that you have a Point class which represents any point with (x, y) coordinates. Where X, Y are both integers in this class. You will overload some operators to work with object of this class. You have to overload the operator.

The operators are:

a. (>>) operator, to input (X, Y) values from keyboard.

b. (<<) operator, to output (X, Y) values.

c. (+=) operator, to add an integer value to (X, Y) values.

d. (==) operator, to test if two points are equal or not.

e. (!) Operator, to swap (X, Y) values to be (Y, X).

f. (/) operator, to find the distance between two points.

**Question#5 Operator Overloading**

Use friend function to write a program for employee salary of Sitara Private Limited.

Make a class and define following data members:

* Name of Employee
* Rank of employee (Basic pay scale)
* Basic pay
* MD (Medical allowance)
* HR (House Rent)
* Gross pay (total pay)

**Task 1:** Define a private Member function of class

* Display ( );

Display complete record of employee including Name of employee, rank, basic pay and gross

pay and Next year pay using this pointer.

**Task 2:** Also define following functions

* Read\_record ( ); (friend function)

To take record of employee like Name of employee, Rank, basic pay from user.

* Gross\_pay ( ); (friend function)

Calculate the gross pay of employee using basic pay and allowances.

* Annual\_increment ( ); (Friend function)

In the next year the pay of employee increases 20% of basic pay.

**Task 3:** Create another friend function through which you’ll call Display() defined in Task1.

Note:

* Medical allowance value is 60% of basic pay and House rent have 28.9% of basic pay.
* Write a constructor in which data members should be initialized and display message “Constructor”.
* Use efficient data type (string, int, float etc.)

At the end define destructor and display message “Software Developed by {your\_name}”.

**Question#6 Operator Overloading**

You are required to implement a **Simplex** transmission mode. In Simplex mode, the

communication is unidirectional, as on a one-way street. Only one of the two devices on a link

can transmit, the other can only receive. A **possible** solution can be designed as:

Write a class **Channel** having private attributes int frequency, time and date (see **time\_t** data

type), string port having public member functions inputDetails(), printDetails().

Write a struct **message** having string mess, int mess\_id.

Write a class **Sender** having private attributes an object of struct message, an object of class

Channel and public member function LoadMessage() which inputs the message, channel details

and generates a random mess\_id.

Write a class **Receiver** having private attributes an object of struct message, bool received and

public member function printMessage() which prints the mess\_id and message.

Now, the problem is that these two classes can not communicate with each other directly.

Hence, we use a mutual friend function called **path()** (can only have 2 parameters) which will

assign the Sender’s message to Receiver’s message.

Your task is to create a menu driven program having options

1. Load Message (channel details and message will be inputted after this selection)

2. Print Message (the path() will be called and then message will be printed with the

details of the channel)

3. Exit

Your main() will have only two objects of class Sender and Receiver and only functions told

above, you can only make another friend function if required.

**Question#7 Operator Overloading**

**Task1:**

Write a class named “Base” having the following integer attributes as:

1. Base1
2. Base2
3. Base3 (private)

Write a default constructor for Base that should initialize all the attributes to 0. Write an

overloaded constructor that should initialize all the attributes to the passed integer. Write a

display method to display the attributes of Base class.

**Task2:**

Write a class Friend\_Class. Make it a friend of Base class.

Write a method AccessMethod in Friend\_Class class.

Access the members of Base class in this method (public as well as private).

**Task3:**

Write a class Friend2\_Class. Make it a friend of Friend\_Class.

Write a method AccessMethod in this Friend2\_Class class.

Access the members of Friend\_Class in this method and also try to access the members of Base

class (check whether you can access its public or private members).