

National University of Computer and Emerging Sciences



Lab Manual Object Oriented Programming

Course Instructor	Ms. Hafsa Tariq
Lab Instructor (s)	Ms. Yusra Arshad Ms. Sonia Anum
Section	BSCS – 2J
Semester	Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Friend Function and Friend Classes
- ✓ Unary and Binary Operator Overloading

Task 01: Friend Functions

Q no.1:

Create two classes **DM** and **DB** that store the value of distances. DM stores distance in *Meters* and *centimeters* and DB in *feet* and *inches*.

Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a *friend* function to carry out the addition operation. The object that stores the results maybe a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feed and inches or meters and centimeters depending on the object on display.

Q no.2:

Write a program with a class **Integer** that contains an array of integers. Initialize the integer array in the constructor of the class. Then create *friend* functions to the class

- Find the largest integer in the array.
- Find the smallest integer in the array.
- Find the repeated elements in array.
- Sort the elements of array in ascending order.
- Create a destructor that sets all of the elements in the array to 0.

Task 02: Operator Overloading

Q no. 1:

A polynomial $P1(x) = x^4 + 2x^2 + 5$ has three terms: x^4 , $2x^2$ and 5. Coefficients of these terms are 1, 2 and 5 respectively while exponents are 4, 2 and 0 respectively. To work with Polynomials, a definition of class Polynomial is given below and memory configuration for P1 is shown as follows:

<pre>class Polynomial { private: int totalTerms;//Total terms in a Polynomial int* coeff;//to save array of coefficients int* exp; //to save array of exponents };</pre>	<table><tr><td>totalTerms</td><td>3</td><td></td><td></td><td></td></tr><tr><td>coeff</td><td></td><td>1</td><td>2</td><td>5</td></tr><tr><td>exp</td><td></td><td>4</td><td>2</td><td>0</td></tr></table> <p>P1(x)</p>	totalTerms	3				coeff		1	2	5	exp		4	2	0
totalTerms	3															
coeff		1	2	5												
exp		4	2	0												

Your task is to complete the definition of Polynomial class such that the main program runs successfully. Make sure that your program doesn't consume extra memory space and it should not leak any memory.

```

void main()
{
    int coeff_P1[] = {1,2,5}; //Coefficients for Polynomial P1

    int exp_P1[] = {4,2,0}; //Exponents for Polynomial P1

    int coeff_P2[] = {4,3}; //Coefficients for Polynomial P2
    int exp_P2[] = {6,2};    //Exponents for Polynomial P2

    Polynomial P1(3, coeff_P1, exp_P1); //Creates P1 with 3 terms (P1 = 1x^4 + 2x^2 + 5x^0 )
    Polynomial P2(2, coeff_P2, exp_P2); //Creates P2 with 2 terms (P2 = 4x^6 + 3x^2)

    cout<<"P1 = "<<P1<<endl; //Prints P1 = x^4+2x^2+5
    cout<<"P2 = "<<P2<<endl; //Prints P2 = 4x^6+3x^2

    if(!P1)
        cout<<"P1 is zero"<<endl; /*if polynomial has only 1 term and its coeff and exp are zero. i.e. if p1 = 0.*/
    if(P1 != P2)
        cout<<"P1 is Not Equal to P2"<<endl;
    cout<<++P1<<endl; //adds 1 in all the coefficient.
    cout<<P1<<endl;
    cout<<P1++<<endl; //adds 1 in all the coefficient.
    cout<<P1<<endl;
    Polynomial P3 = P1+P2; //Adds P1 and P2 and saves result in P3. You may consume extra
    //space for resultant Polynomial in Add function
    cout<<"P3 = "<<P3<<endl; //Prints P3 = 4x^6+x^4+5x^2+5

    P3 = 2 + P1; //Assume P1 already has a constant term, add 2 in it.
    cout<<"P3 = "<<P3<<endl;
}

```

END

