# National University of Computer and Emerging Sciences



**Lab Manual 05**

# Object Oriented Programming

|  |  |
| --- | --- |
| Course Instructor | Mr. Waqas Manzor |
| Lab Instructor (s) | Hira Ilyas  Sidra Zafar |
| Section | BSE- A |
| 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*.

1 foot= 12 inches

1 inch=2.54 cm

For example

Distance1=4 m 15 cm

Distance1=5 ft 10 inch

T\_cm= (4\*100+15)+ ((5\*12)+10)\*2.54

M=t\_cm/100

Cm=t\_cm%100

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 would be a DM object

* Create parameterized constructors of DM and DB such that DM(int m, int cm) and DB(int ft, int inches)
* Create a friend function *friend DB add(DB, DM)* which returns an object of DB after performing the addition on DB and DM objects.

**Q no.2:**

Write a program with a class **Integer** that contains an array of integers. Initialize the integerarray 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.

**Task 02: Operator Overloading**

**Q no. 1:**

Implement a class called **Complex**. The Complex class will have two data members:

* int real; // The real part of complex number
* int imaginary; // Imaginary part of the complex number.

You have to implement default constructor, overloaded constructor, copy constructor, destructor and overload the operators **+, - , << , >>, ==, !=,=**

**Sample Run:**

|  |  |
| --- | --- |
| **Driver.cpp** | **Output** |
| int main()  {  Complex C1;  Complex C2(5,6);  Complex C3;  cout<<"Input a complex number"<<endl;  cin>>C3;  cout<<C1<<C2<<C3;  if(C1==C2)  cout<<"C1 == C2"<<endl;  else  cout<<"C1 != C2"<<endl;  if(C1!=C3)  cout<<"C1 != C3"<<endl;  else  cout<<"C1==C3"<<endl;  Complex C4= C2 - C3;  cout<<C4;  system("pause");  return 0;  } |  |

**Note:**

* Deallocate all dynamically allocated memory.
* Make separate complex.h, complex.cpp and driver.cpp files.
* Follow all the code indentation, naming conventions and code commenting guidelines.

**Q no. 2**

A polynomial P1(x) = x4+2x2+5 has three terms: x4, 2x2 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:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **class Polynomial**  **{**  **private:**  int totalTerms;//Total terms in a Polynomial  int\* coeff;//to save array of coefficients  int\* exp; //to save array of exponents  **};** | |  |  |  | | --- | --- | --- | | totalTerms | 3 |  | | coeff |  | |  |  |  | | --- | --- | --- | | 1 | 2 | 5 | | | exp |  | |  |  |  | | --- | --- | --- | | 4 | 2 | 0 | | |  | **P1(x)** |  | |

**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