National University of Computer and Emerging Sciences

Lab Manual # 06

Object oriented programming

Course Instructor	Ms. Arooj Khalil
Lab Instructor(s)	Mamoona Akbar Saleha Batool
Section(s)	BSE-2B1 BSE-2B2
Semester	Spring 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- Operator overloading
- Inline function
- Friend function

TASK 1:

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

- int a;
- int b;

You have to implement the following:

1. Implement all getters/setters.
2. Write an overloaded and default constructor.
3. Implement following member functions such that they are inline.

```
void sum();
void difference();
void product();                    // Write in Comments section what are the
void division();                      benefits of inline function.
```

4. Write a suitable main() function to show desired functionality.

TASK 2:

Implement a class called Quadratic. The class will have three data members:

- int a; // First part of quadratic equation
- int b; // Second part of the equation
- int c; //Third part of quadratic equation.

//It'll form a number as ax^2+bx+c

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

- + => Add 2 quadratic objects (**Member as well as friend function)**
- **-** => Subtract one quadratic object from other **(Member as well as friend function)**
- * => Multiply a constant with Quadratic object **(Member Function only)**
- >> => Input a quadratic object **(Friend Function)**
- << => Output a quadratic object (**Friend Function)**
- == => Equality Operator **(Member Function only)**
- **!=** => In-equality operator **(Member Function only)**
- = => Assignment operator **(Member Function only)**

<u>TASK 2:</u>
Implement following **ComplexNumber** class and write driver program to test the implementation. Do not change the class definition in any way or form.

```
class ComplexNumber
{
private:
        int real;
        int imaginary;
        static int count;  //will count the total number of objects created
public:
        ComplexNumber(int, int); //with default arguments
        ~ComplexNumber(); //Does Nothing.
        void Input();
        void Output();
        static int countDisplay; //will return the total number of by incrementing as the object is created
        bool IsEqual(ComplexNumber);
        ComplexNumber Conjugate();
        // Adding two complex numbers ( a + bi ) and ( c + di ) yields ( (a+b) + (c+d)i )
        ComplexNumber  operator+ (const ComplexNumber & num);
        //Subtracting two complex numbers (a + bi) and (c + di) yields ((a-b) + (c-d)i).
        ComplexNumber  operator- (const ComplexNumber & num);
        //Multiplying two complex numbers(a + bi)and(c + di) yields ((ac-bd) + (ad+bc)i).
        ComplexNumber  operator* (const ComplexNumber & num);
        //Increment and decrement operators should only add 1 or subtract 1 from real part
        ComplexNumber  & operator ++(); // pre-increment
        ComplexNumber  & operator --(); // pre-decrement
        ComplexNumber  operator ++(int); // post-increment
        ComplexNumber  operator --(int); // post-decrement
        bool operator>=(const ComplexNumber& num);
        bool operator<=(const ComplexNumber& num);
        bool operator!=(const ComplexNumber& num);
};
```