Modern Education Society's

College of Engineering, Pune

NAME OF STUDENT: Prathamesh Kalyan Sable	**CLASS: SE Comp 1**
SEMESTER/YEAR: Sem – 3 / 2022	**ROLL NO: F21111015**
DATE OF PERFORMANCE: 12/09/2022	**DATE OF SUBMISSION: 19/09/2022**
EXAMINED BY: Mrs. R. H. Shende	**EXPERIMENT NO: Group A - 01**

Title: Complex Number

Objectives: To understand operator overloading, constructor and data hiding.

Problem Statement:

Implement a class Complex which represents the Complex Number data type. Implement thefollowing operations:

1. Constructor (including a default constructor which creates the complex number 0+0i).
2. Overloaded **operator+** to add two complex numbers.
3. Overloaded **operator*** to multiply two complex numbers.
4. Overloaded << and >> to print and read Complex Numbers.

Outcomes: To understand operator overloading concept.

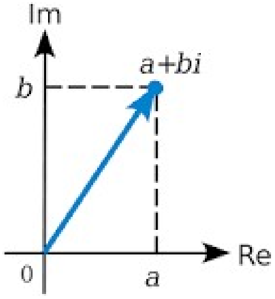
Software Requirements: ubuntu Operating system, Eclipse and Gedit.

Hardware Requirements: P4 System.

Prerequisite: Object oriented concepts.

2.1 Theory:

A **complex number** is a **number** that can be expressed in the form a + bi, where a and b are real**numbers** and i is the imaginary unit, that satisfies the equation $i^2 = -1$. In this expression, a is thereal part and b is the imaginary part of the **complex number**.



Operations on Complex number:

1. **Addition of Complex Numbers** Addition of two complex numbers a + b i and c + d i is defined as follows.

   (a+b i) + (c+d i) = (a + c) + (b + d) i

   This is similar to grouping like terms: real parts are added to real parts and imaginary parts areadded to imaginary parts.

2. **Subtraction of Complex Numbers**

   The subtraction of two complex numbers a + b i and c + d i is defined as follows. (a + b i) - (c + d i) = (a - b) + (b - d) i

3. **Multiply Complex Numbers**

   The multiplication of two complex numbers a + b i and c + d i is defined as follows. (a + b i) (c + d i) = (a c - b d) + (a d + b c) i
   However, you do not need to memorize the above definition as the multiplication can be carried out using properties similar to those of the real numbers and the added property i ^{2} = -1. (See the example below)

2.2 OOP Features & Concepts used in this practical:

Operator Overloading: You can redefine or overload most of the built-in operators available in C++. Thus, a programmer can use operators with user-defined types as well.

Overloaded operators are functions with special names the keyword operator followed by the symbol for the operator being defined. Like any other function, an overloaded operator has a return type and a parameter list.

Box operator+ (const Box&);

declares the addition operator that can be used to **add** two Box objects and returns final Boxobject.

Overloadable/Non-overloadable Operators:

Following is the list of operators which can be overloaded:

+	-	*	/	%	^
&	|	~	!	,	=

<	>	<=	>=	++	--
<<	>>	==	!=	&&	||
+=	-=	/=	%=	^=	&=
|=	*=	<<=	>>=	[ ]	( )
->	->*	new	new [ ]	delete	delete [ ]

Operator that are not overloaded are follows

scope operato - ::

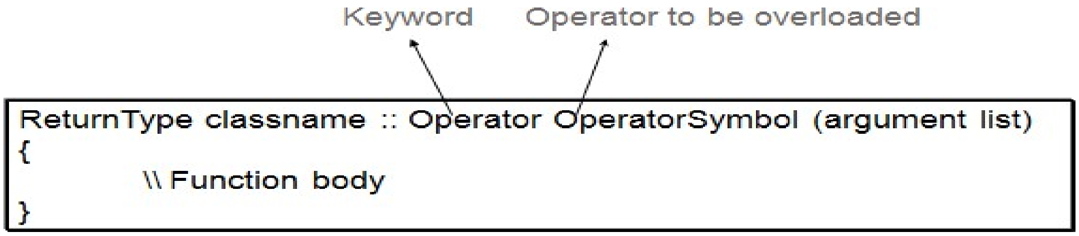
sizeof

member selector - .

member pointer selector - *

ternary operator - ?:

Syntax of operator overloading:



```
ReturnType classname :: Operator OperatorSymbol (argument list)
{
        \\ Function body
}
```

For calling function:

resultant = Object operatorsymbol Object;

resultant will vary depending upon operation done and return type of function.

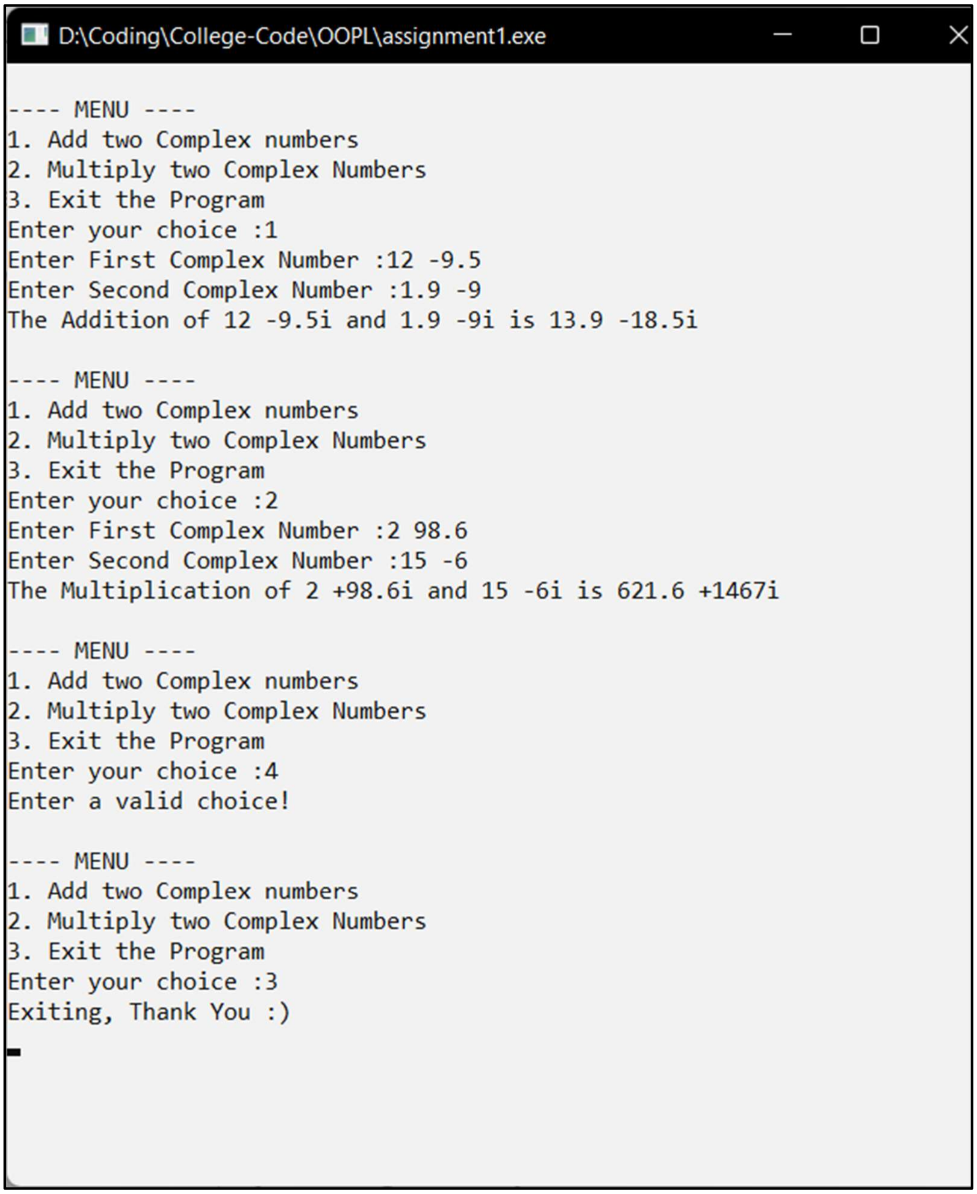
2.3 Conclusions: Complex number operations are implemented with the operator overloadingconcept

Source Code:

```
#include <iostream>
using namespace std;

class Complex{
public:
    float real, imag;
    // default constructor
    Complex(){
        real = 0;
        imag = 0;
    }
    Complex(float real, float imag){
        this->real = real;
        this->imag = imag;
    }
    // to display complex number
    void disp(){
        cout << *this;
    }
    // addition operator overloading
    Complex operator+(Complex num2){
        Complex res;
        res.real = (real + num2.real);
        res.imag = (imag + num2.imag);
        return res;
    }
    // multipication operator overloading
    Complex operator*(Complex num2){
        Complex res;
        res.real = ((real * num2.real) - (imag * num2.imag));
        res.imag = ((real * num2.imag) + (imag * num2.real));
        return res;
    }

    // insertion and extraction overloading using friend function
    friend ostream &operator<<(ostream &object, Complex &num);
    friend istream &operator>>(istream &object, Complex &num);
};
ostream &operator<<(ostream &object, Complex &num){
    // displaying complex number
    if (num.imag < 0){
        object << num.real << " " << num.imag << "i";
    }else{
        object << num.real << " +" << num.imag << "i";
    }
    return object;
}
```

```
istream &operator>>(istream &object, Complex &num){
    // real and imag part input
    object >> num.real >> num.imag;
    return object;
}

int main(){
    // Declearing Variables
    bool loop_control = true;
    int choice;
    Complex num1, num2, res;
    while (loop_control){
        cout << "\n---- MENU ----" << endl;
        cout << "1. Add two Complex numbers \n2. Multiply two Complex Numbers
\n3. Exit the Program \nEnter your choice :";
        cin >> choice;
        switch (choice){
        case 1:
            cout << "Enter First Complex Number :";
            cin >> num1;
            cout << "Enter Second Complex Number :";
            cin >> num2;
            res = num1 + num2;
            cout << "The Addition of " << num1 << " and " << num2 << " is " <<
res << endl;
            break;
        case 2:
            cout << "Enter First Complex Number :";
            cin >> num1;
            cout << "Enter Second Complex Number :";
            cin >> num2;
            res = num1 * num2;
            cout << "The Multiplication of " << num1 << " and " << num2 << "
is " << res << endl;
            break;
        case 3:
            loop_control = false;
            cout << "Exiting, Thank You :)" << endl;
            break;
        default:
            cout << "Enter a valid choice!" << endl;
            break;
        }
    }
    return 0;
}
```

OUTPUT:



```
---- MENU ----
1. Add two Complex numbers
2. Multiply two Complex Numbers
3. Exit the Program
Enter your choice :1
Enter First Complex Number :12 -9.5
Enter Second Complex Number :1.9 -9
The Addition of 12 -9.5i and 1.9 -9i is 13.9 -18.5i

---- MENU ----
1. Add two Complex numbers
2. Multiply two Complex Numbers
3. Exit the Program
Enter your choice :2
Enter First Complex Number :2 98.6
Enter Second Complex Number :15 -6
The Multiplication of 2 +98.6i and 15 -6i is 621.6 +1467i

---- MENU ----
1. Add two Complex numbers
2. Multiply two Complex Numbers
3. Exit the Program
Enter your choice :4
Enter a valid choice!

---- MENU ----
1. Add two Complex numbers
2. Multiply two Complex Numbers
3. Exit the Program
Enter your choice :3
Exiting, Thank You :)
```