

SE(ECE)	Data Structures and Algorithms	
Experiment No: 1	Program to demonstrate basic C++ concepts — classes, objects, constructors, and operator overloading for complex number arithmetic (Addition, Subtraction, Multiplication)	Page: 1/7

Aim

To implement a C++ program that demonstrates the use of classes, objects, constructors, and operator overloading for performing arithmetic operations (Addition, Subtraction, Multiplication) on complex numbers.

Objectives

1. To understand the concept of classes and objects in C++.
2. To learn the use of constructors for initializing objects.
3. To apply the concept of operator overloading for user-defined operations.
4. To perform arithmetic operations (Addition, Subtraction, Multiplication) on complex numbers using operator overloading.

Theory

Class

A class in C++ is a user-defined data type that acts as a blueprint for creating objects.

It groups:

- **Data members** (variables that hold data)
- **Member functions** (methods that operate on the data)

A **class is** like a blueprint or *template*, it defines *what* an object will contain and *what it can do*, but it doesn't occupy memory until objects are created.

Object

An object is an instance of a class.

When you create an object, memory is allocated for its data members, and you can use it to access member functions.

Analogy:

- Class = Blueprint of a house
- Object = Actual house built using the blueprint

Encapsulation

Encapsulation is the process of wrapping data and member functions together inside a class and controlling access to the data.

Encapsulation protects the data of an object from unauthorized or direct external access. This is achieved through the use of access specifiers (private, protected, public):

SE(ECE)	Data Structures and Algorithms	
Experiment No: 1	Program to demonstrate basic C++ concepts — classes, objects, constructors, and operator overloading for complex number arithmetic (Addition, Subtraction, Multiplication)	Page: 2/7

- **private** members: These are accessible only from within the class itself. This is where sensitive data is typically declared.
- **public** members: These are accessible from outside the class. Public member functions often act as the interface for interacting with the private data, using "getter" and "setter" methods.
- **protected** members: These are accessible within the class and by derived classes.

Encapsulation ensures that data is accessed and modified only through the defined public interface (member functions). This provides control over how the data is manipulated, allowing for validation, error checking, and maintaining data integrity.

Constructor in C++

A constructor is a special member function of a class that is automatically invoked when an object of that class is created.

Its primary purpose is to initialize the object's data members and establish its initial state.

Key characteristics of C++ constructors:

- **Same Name as Class:** A constructor must have the exact same name as the class it belongs to.
- **No Return Type:** Constructors do not have a return type, not even void.
- **Automatic Invocation:** They are called automatically by the compiler when an object of the class is instantiated.
- **Initialization:** Constructors are responsible for initializing the member variables of the newly created object.
- **Overloading:** Constructors can be overloaded, meaning a class can have multiple constructors with different parameter lists.
- **Public Access:** Constructors are typically declared in the public section of the class.

Operator Overloading

- C++ allows redefining the behaviour of operators for user-defined data types.
- By overloading operators, objects of a class can be manipulated using operators like +, -, *.
- Example: For a class Complex, we can overload + to add two complex numbers.

Complex Number Arithmetic

A complex number is of the form: $z=a+ib$

where

a = real part

b = imaginary part

Operations:

Addition: $(a+ib)+(c+id)=(a+c)+i(b+d)$

Subtraction: $(a+ib)-(c+id)=(a-c)+i(b-d)$

Multiplication: $(a+ib) \times (c+id)=(ac-bd)+i(ad+bc)$

SE(ECE)	Data Structures and Algorithms	
Experiment No: 1	Program to demonstrate basic C++ concepts — classes, objects, constructors, and operator overloading for complex number arithmetic (Addition, Subtraction, Multiplication)	Page: 3/7

Algorithm

Algorithm for Addition

1. Start.
2. Define a class Complex with data members real and imag.
3. Define a constructor to initialize real and imaginary parts.
4. Overload the + operator to add two complex numbers.
 - o Add real parts.
 - o Add imaginary parts.
 - o Return the result as a Complex object.
5. In main(), create two objects and use + operator.
6. Display result.
7. Stop.

Algorithm for Subtraction

1. Start.
2. Overload the - operator inside Complex class.
3. Subtract real parts and imaginary parts.
4. Return the result as a Complex object.
5. In main(), create two objects and use - operator.
6. Display result.
7. Stop.

Algorithm for Multiplication

1. Start.
2. Overload the * operator inside Complex class.
3. Compute multiplication using:

$$(a+ib) \times (c+id) = (ac - bd) + i(ad + bc)$$
4. Return result as a Complex object.
5. In main(), create two objects and use * operator.
6. Display result.
7. Stop.

SE(ECE)	Data Structures and Algorithms	
Experiment No: 1	Program to demonstrate basic C++ concepts — classes, objects, constructors, and operator overloading for complex number arithmetic (Addition, Subtraction, Multiplication)	Page: 4/7

Applications

1. Used in scientific and engineering computations involving complex numbers.
2. Applicable in electrical engineering for analyzing AC circuits (impedance, phasors).
3. Useful in signal processing and control systems.
4. Forms the basis of polynomial equation solving in mathematics.

References

1. Yedidyah Langsam, Moshe J Augenstein, Aaron M Tenenbaum – Data structures using C and C++ - PHI Publications (2nd Edition).
2. Ellis Horowitz, Sataraj Sahni- Fundamentals of Data Structures – Galgotia Books source.

Questions

1. Which of the following best defines a class in C++?

- a) A variable that stores multiple values
- b) A collection of functions only
- c) A user-defined data type containing data members and member functions
- d) A function that initializes objects

2. An object in C++ is:

- a) A blueprint for creating a class
- b) An instance of a class
- c) A function inside a class
- d) A constructor

3. Which of the following statements about constructors is correct?

- a) Constructors must have a return type
- b) Constructors have the same name as the class
- c) Constructors can be called explicitly only
- d) A class cannot have more than one constructor

4. Which type of constructor is invoked when an object is created without arguments?

- a) Copy constructor
- b) Default constructor
- c) Parameterized constructor
- d) Destructor

SE(ECE)	Data Structures and Algorithms	
Experiment No: 1	Program to demonstrate basic C++ concepts — classes, objects, constructors, and operator overloading for complex number arithmetic (Addition, Subtraction, Multiplication)	Page: 5/7

5. What is operator overloading in C++?

- a) Redefining the meaning of existing operators for user-defined data types
- b) Using operators for built-in data types
- c) Writing multiple operators in a single function
- d) Using the same operator in multiple languages

6. Which operator cannot be overloaded in C++?

- a) +
- b) -
- c) =
- d) ::

7. Which of the following is true about operator overloading in C++?

- a) Operator overloading creates new operators in C++
- b) Operator overloading only works with built-in data types
- c) Operator overloading gives additional meaning to existing operators
- d) All operators must be overloaded

8. Which of the following is a special member function in C++?

- a) Destructor
- b) Constructor
- c) Operator overload function
- d) All of the above

9. In operator overloading, which keyword is used inside the class?

- a) class
- b) friend
- c) operator
- d) this

10. Given the statement:

Complex c3 = c1 + c2;

What must be done inside the Complex class?

- a) Overload the - operator
- b) Overload the + operator
- c) Create a copy constructor
- d) Define a destructor

SE(ECE)	Data Structures and Algorithms	
Experiment No: 1	Program to demonstrate basic C++ concepts — classes, objects, constructors, and operator overloading for complex number arithmetic (Addition, Subtraction, Multiplication)	Page: 6/7

11. Which access specifier allows class members to be accessed only within the class itself?

- a) public
- b) private
- c) protected
- d) static

12. What happens if a constructor is not defined in a class?

- a) Compilation error occurs
- b) The program crashes at runtime
- c) Compiler provides a default constructor automatically
- d) The class cannot be instantiated

13. Which of the following is true about destructors in C++?

- a) They can take parameters
- b) They are overloaded like constructors
- c) They have the same name as the class, prefixed with ~
- d) They return an integer value

14. Which keyword is used to define a friend function that can access private members of a class?

- a) private
- b) protected
- c) friend
- d) static

15. Consider the code:

Complex operator+(Complex c);

What does this function declaration represent?

- a) A default constructor
- b) A destructor
- c) Overloading the + operator for the Complex class
- d) Function overloading

SE(ECE)	Data Structures and Algorithms	
Experiment No: 1	Program to demonstrate basic C++ concepts — classes, objects, constructors, and operator overloading for complex number arithmetic (Addition, Subtraction, Multiplication)	Page: 7/7

Conclusion

A	C	O	T	Sign.