
# Object Oriented Programming in Java Laboratory

**Name of Student: Rishabh Jain**
**Roll Number**: SE21IT047
**PRN No:72139725F**
**Batch: A3**
**Lab Coordinators :(**Mrs. Sapna Kolambe)

**Assignment No: -01**

**Department: -Information Technology**

**Batch: -A2**

**Roll Number:** Second Year-IT047

**Date of Assignment: -**

**Date of Completion: -**

# Assignment No 01

**Title:**

**Aim:**

**Design a class 'Complex 'with data members for real and imaginary part. Provide default and Parameterized constructors. Write a program to perform arithmetic operations of two complex numbers.**

**Objective:**

**Differentiate various programming paradigms. Identify classes, objects, methods, and handle object creation, initialization, and destruction to model real-world problems. To learn the concept of class, object and constructor**

**Outcomes:**

1. Students will be able to understand basic concept of class and object in Java.

2. Students will be able to demonstrate different operations on class and object.

3. Students will be able to demonstrate different Operations on any object by using own logic.

**Hardware Requirement:** Any CPU with i3 Processor or similar, 1 GB RAM or more,2 GB Hard Disk or more

**Software Requirements:** 32/64-bit Linux (Ubuntu/Fedora) Operating System, latest any Java Tool Like NetBeans, Eclipse, or any Online Java Compiler.

**Theory:**

An Object is an instance of a Class. A class is like a blueprint while an instance is a copy of the class with *actual values*.
An object consists of:

- **State:** It is represented by the attributes of an object. It also reflects the properties of an object.
- **Behaviour:** It is represented by the methods of an object. It also reflects the response of an object to other objects.
- **Identity:** It gives a unique name to an object and enables one object to interact with other objects.

**Declaring Objects (Also called instantiating a class)**
When an object of a class is created, the class is said to be instantiated. All the instances share the attributes and the behaviour of the class. But the values of those attributes, i.e., the state are unique for each object. A single class may have any number of instances.

Also, we have used a do while loop to repeat the whole function again and thus it will again ask the user to enter another set of operation.

If the user wants to exit the operation, he/she can enter the $5^{th}$ option to exit the set of operation.

**Theory**:
1.What is a class in Java?
A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.
A class in Java can contain:
✓ Fields
✓ Methods
✓ Constructors
✓ Blocks
✓ Nested class and interface
Syntax to declare a class:

```
class
{
field;
method;
```

}

2. Instance variable in Java

➢ A variable which is created inside the class but outside the method is known as an instance variable. Instance variable doesn't get memory at compile time. It gets memory at runtime when an object or instance is created. That is why it is known as an instance variable.

➢ Method in Java

➢ In Java, a method is like a function which is used to expose the behaviour of an object.

➢ new keyword in Java

➢ The new keyword is used to allocate memory at runtime. All objects get memory in Heap memory area. classname object =new classname();

3.What is an object in Java ?

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible).

An object has three characteristics:

State: represents the data (value) of an object.

Behavior: represents the behaviour (functionality) of an object such as deposit, withdraw, etc.

Identity: An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

For Example, Pen is an object. Its name is Reynolds; color is white, known as its state. It is used to write, so writing is its behaviour.

An object is an instance of a class. A class is a template or blueprint from which objects are created.

Object Definitions:

An object is a real-world entity.
An object is a runtime entity.
The object is an entity which has state and behavior.
The object is an instance of a class.

Object and Class Example: main within the class
In this example, create a Student class which has two data members id and name. Create the object of the Student class by new keyword and print the object's value.

**********

Here, create a main() method inside the class.
File: Student.java

```
//Java Program to illustrate how to define a class and fields
//Defining a Student class.
class Student
{
//defining fields
int id;//field or data member or instance variable
String name;
//creating main method inside the Student class
public static void main (String args[])
{
//Creating an object or instance
Student s1=new Student();
//creating an object of Student
//Printing values of the object
System.out.println(s1.id);
//accessing member through reference variable
System.out.println(s1.name);
}
}
```

**********

Object and Class Example: main outside the class
In real time development, create classes and use it from another class. It is a better approach than previous one. Let's see a simple example, where main () method in defined in another class.

File: TestStudent1.java

```
//Java Program to demonstrate having the main method inanother class
//Creating Student class.
class Student
{
int id;
String name;
}
//Creating another class TestStudent1 which contains the main method
class TestStudent1
{
 public static void main(String args[])
{
Student s1=new Student();
System.out.println(s1.id);
System.out.println(s1.name);
}
}
```

4.Constructors in Java In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory. It is a special type of method which is used to initialize the object. Every time an object is created using the new() keyword, at least one constructor is called. It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

There are two types of constructors in Java: no-arg constructor, and parameterized constructor.

It is not necessary to write a constructor for a class. It is because java compiler creates a default constructor if your class doesn't have any.

•Rules for creating Java constructor

There are two rules defined for the constructor.

1. Constructor name must be the same as its class name
2. A Constructor must have no explicit return type
3. A Java constructor cannot be abstract, static, final, and synchronized

•Types of Java constructors

There are two types of constructors in Java:

1. Default constructor (no-arg constructor)
2. Parameterized constructor

Java Default Constructor

A constructor is called "Default Constructor" when it doesn't have any parameter.

Syntax of default constructor:

{ }

Example of default constructor
In this example, creating the no-arg constructor in the Bike class. It will be invoked at the time of object creation.

```
//Java Program to create and call a default constructor
class Bike1
{
//creating a default constructor Bike1()
{
System.out.println("Bike is created");
}
//main method
public static void main(String args[])
{
//calling a default constructor
        Bike1 b=new Bike1();
}
}
```

Java Parameterized Constructor
A constructor which has a specific number of parameters is called a parameterized constructor.
The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.
Syntax of parameterized constructor:

```
class<class_name>
{
int x;
float y;
class<class_name> (int m, float n)
{
x=m;
y=n;
}
```

Example of parameterized constructor
In this example, create the constructor of Student class that have two parameters. Initialize this parameter value using parameterised constructor.

//Java Program to demonstrate the use of the parameterized constructor.

```
class Student4
{
int id;
String name;
Student4(int i,String n)//creating a parameterized constructor
{
        id = i;
        name = n;
}
void display()
{
System.out.println(id+" "+name);
}
public static void main(String args[])
{
Student4 s1 = new Student4(111,"Karan"); //creating objects and passing values
Student4 s2 = new Student4(222,"Aryan");
s1.display(); //calling method to display the values of object
s2. display();
}
}
```

Example: Create a class called Complex for performing arithmetic on complex numbers.
Complex numbers have the form a+bi where a is real part and b is imaginary part and i=√-1.

Use floating point variables to represent the private data of the class.
Provide constructor that enable an object to be initialized when it is declared.
Provide no argument constructor with default values in case no initializers are provided.
Provide public methods for addition, subtraction, multiplication and division of complex numbers.
Pass objects of Complex as parameters of the method.

Algorithm:
1. Begin
2. Define a class operation with instance variables real and imag
3. Input the two complex numbers c1=(a+ib) and c2=(c+id)
4. Define the method add (c1, c2) as (a+ib) + (c+id) and stores result in c3
5. Define the method sub (c1, c2) as (a+ib) - (c+id) and stores result in c3
6. Define the method mul (c1, c2) as (a+ib) * (c+id) and store the result in c3 as (ac-bd) +
i(bc+ad)
7. Define the method div (c1, c2) as (a+ib)/(c+id) and stores the quotient c3 as {(ac+bd)/(c2+d2)} +i{(bc-ad)/(c2+d2)}
8. Define the method display () which outputs each result
9. End

**Input and Output Requirements**:
Program reads real and imaginary parts of two complex numbers through keyboard and displays their sum, difference, product and quotient as result.

## Program:

```
import java.util.Scanner;

public class Main {
   public static void main(String args[]) {
      int ch = 0;
```

```
        float num1, num2, answer;
        Complex cal = new Complex();
        Scanner input = new Scanner(System.in);


        do {
           System.out.println("\n1.Addition");
           System.out.println("\n2.Subtraction");
           System.out.println("\n3.Multiplication");
           System.out.println("\n4.Division");
           System.out.println("\n5.Exit");
           System.out.print("\nEnter the choice --> ");
           ch = input.nextInt();
           if(ch>0 && ch <5) {
              System.out.print("\nEnter the first Number\n");
              System.out.print("Real Number :");
              num1 = input.nextInt();
              System.out.print("Imaginary Number : ");
              num2 = input.nextInt();
              Complex First = new Complex(num1, num2);

              System.out.print("\nEnter the Second Number\n");
              System.out.print("Real Number ");
              num1 = input.nextInt();
              System.out.print("Imaginary Number ");
              num2 = input.nextInt();
              Complex Second = new Complex(num1, num2);


             switch (ch) {
              case 1:
                 cal.Addition(First, Second);
                 break;
              case 2:
                 cal.Subtraction(First, Second);
                 break;
              case 3:
                 cal.Multiplication(First, Second);
                 break;
              case 4:
                 cal.Division(First, Second);
                 break;
              default:
                 System.out.println("\n Your Choice of operation is not
available , Try another operation !");
```

```
            }
          }

          else{
             System.out.println("\nYou Have Chosen to exit !!! \n\n");
          }

       }while (ch != 5) ;
    }
    }

    class Complex
{
    private float real,img;
    Complex()
    {
       real=0;
       img=0;
    }
    Complex(float Comp1,float Comp2)
    {
       real=Comp1;
       img=Comp2;
    }
    void   CollectNumbers(){

    }
    void  Addition(Complex C1,Complex C2)
    {
       float real,imag;
    this.real = (C1.real + C2.real);
    this.img = (C1.img + C2.img);
       System.out.println("\nThe Addition of the two numbers is =
"+this.real+" + "+this.img+" i\n" );
    }
    void Subtraction(Complex C1,Complex C2)
    {
       float real,img;
       this.real = (C1.real - C2.real);
       this.img = (C1.img - C2.img);
       System.out.println("\nThe Subtraction of the numbers is =
"+this.real+" + "+this.img+" i\n" );
    }
    void Multiplication(Complex C1,Complex C2)
    {
```

```
        float real,img;
        this.real = (C1.real*C2.real) - (C1.img*C2.img);
        this.img = (C1.real*C2.img) + (C2.real*C1.img);
        System.out.println("\nThe Multiplication of the two numbers is =
    "+this.real+" + "+this.img+" i\n" );
      }
      void Division(Complex C1,Complex C2)
      {
        float real,img,denominator;
        denominator = (C2.real*C2.real) + (C2.img*C2.img);
        this.real = ((C1.real*C2.real) + (C1.img*C2.img))/denominator;
        this.img = ((C2.real*C1.img) - (C1.real*C2.img))/denominator;
        System.out.println("\nThe Division of the two numbers is =
    "+this.real+" + "+this.img+" i\n" );
      }
}
```

## Output:

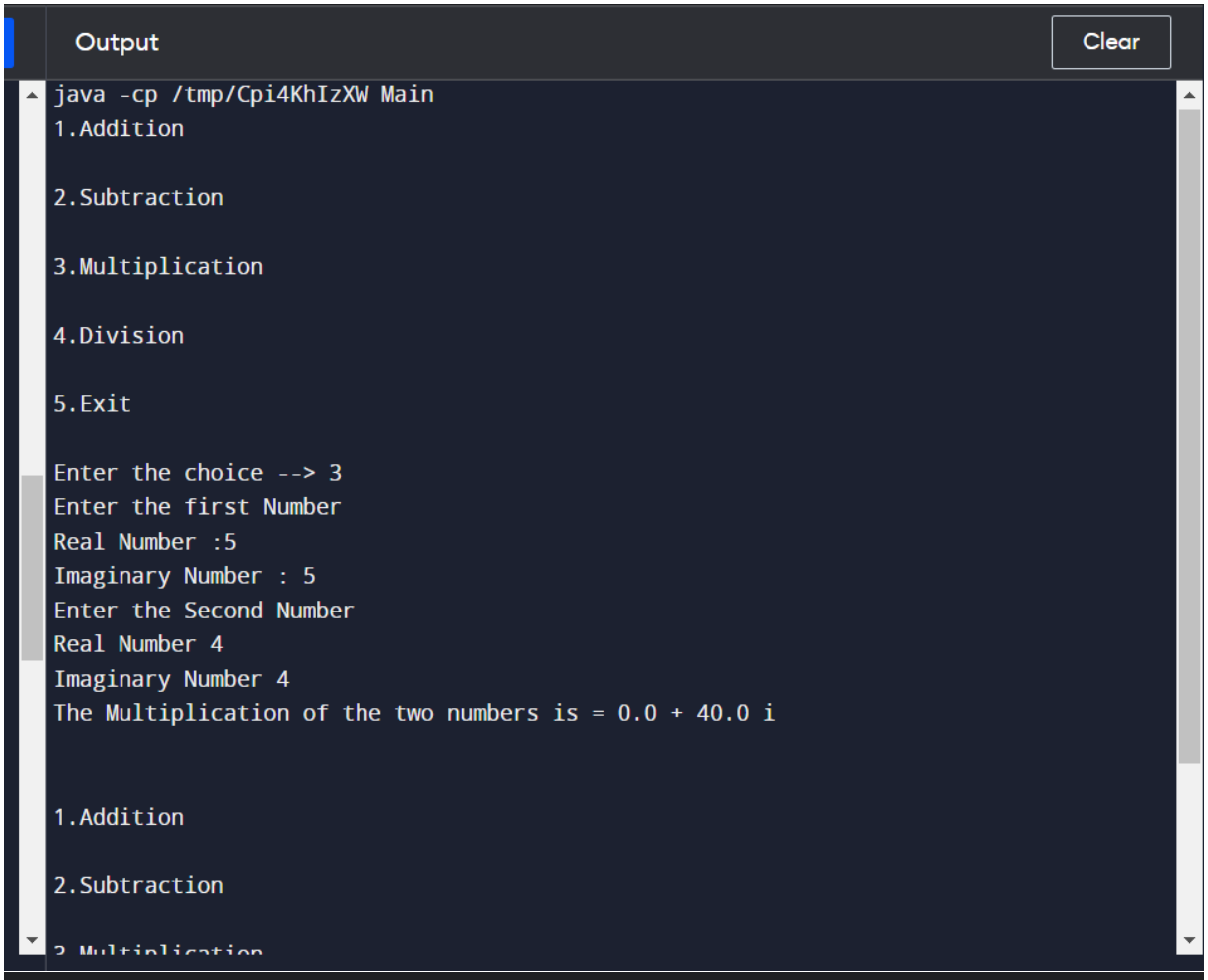
```
1.Addition

2.Subtraction

3.Multiplication

4.Division

5.Exit

Enter the choice --> 4
Enter the first Number
Real Number :20
Imaginary Number : 10
Enter the Second Number
Real Number 5
Imaginary Number 5
The Division of the two numbers is =  3.0 + -1.0 i


1.Addition

2.Subtraction

3.Multiplication
```

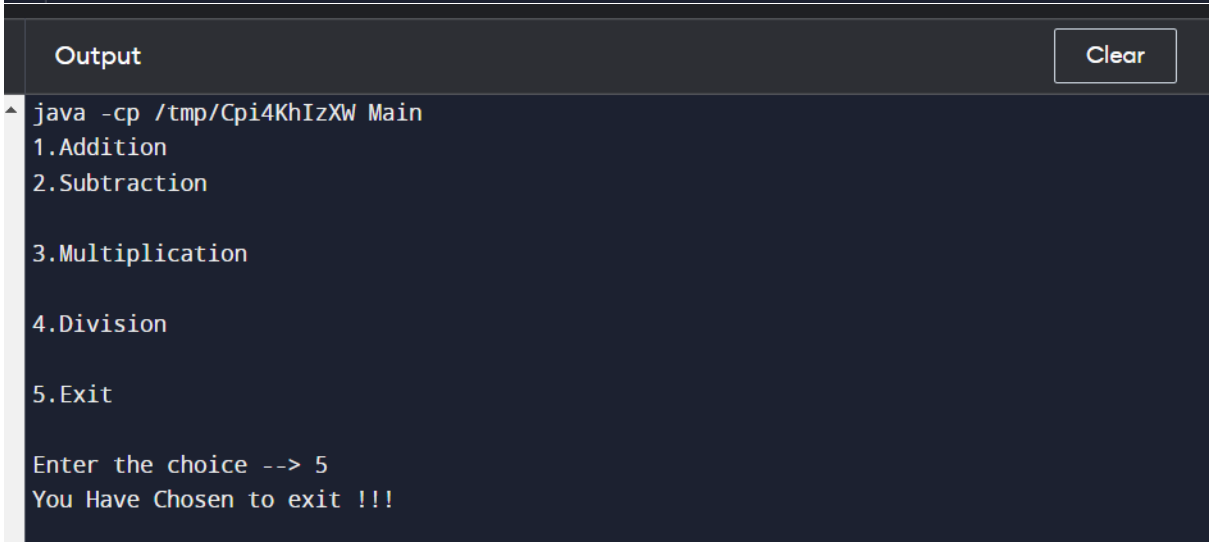
Output Clear

```
java -cp /tmp/Cpi4KhIzXW Main
1.Addition

2.Subtraction

3.Multiplication

4.Division

5.Exit

Enter the choice --> 1
Enter the first Number
Real Number :5
Imaginary Number : 5
Enter the Second Number
Real Number 3
Imaginary Number 3
The Addition of the two numbers is =  8.0 + 8.0 i


1.Addition2.Subtraction

3.Multiplication

4.Division
```

Output Clear

```
java -cp /tmp/Cpi4KhIzXW Main
1.Addition

2.Subtraction3.Multiplication

4.Division

5.Exit

Enter the choice --> 2
Enter the first Number
Real Number :5
Imaginary Number : 5
Enter the Second Number
Real Number 3
Imaginary Number 3
The Subtraction of the numbers is = 2.0 + 2.0 i


1.Addition

2.Subtraction

3.Multiplication

4.Division
```




**Conclusion:**

Thus, in this experiment we have studied and performed different operation on Object Oriented Programming in Java successfully.