**Client.java**

```
mport java.rmi.*;
import java.util.Scanner;



public class Client{
	public static void main(String[] args){
		Scanner sc = new Scanner(System.in);

		try{
			String serverURL = "rmi://localhost/Server";
			ServerIntf serverIntf = (ServerIntf) Naming.lookup(serverURL);

			System.out.print("Enter First Number: ");
			double num1 = sc.nextDouble();

			System.out.print("Enter Second Number: ");
			double num2 = sc.nextDouble();


			System.out.println("First Number Is: " + num1);
			System.out.println("Second Number Is: " + num2);


			System.out.println("--------------- Results ---------------");
			System.out.println("Addition Is: " +serverIntf.Addition(num1, num2));
			System.out.println("Subtraction Is: " +serverIntf.Subtraction(num1, num2));
			System.out.println("Multiplication Is: " +serverIntf.Multiplication(num1, 
num2));
			System.out.println("Division Is: " +serverIntf.Division(num1, num2));


		}catch(Exception e){
			System.out.println("Exception Occurred At Client!" + e.getMessage());
		}

	}

}
```

**Server.java**

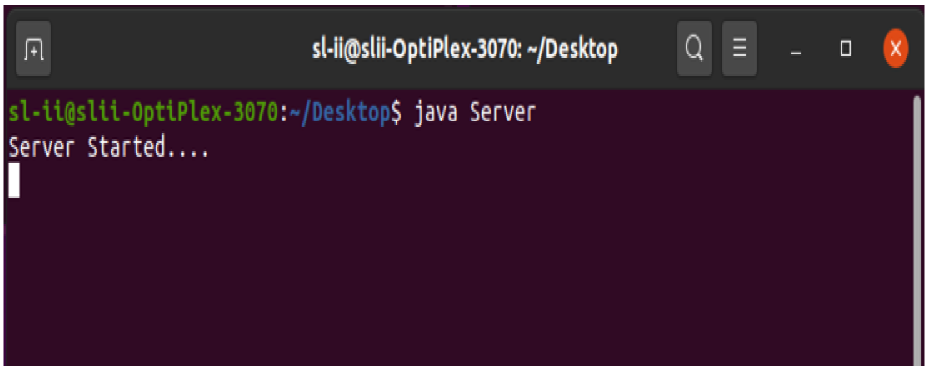
```
import java.rmi.*;

public class Server{

	public static void main(String[] args){

		try{
			ServerImpl serverImpl = new ServerImpl();
			Naming.rebind("Server", serverImpl);

			System.out.println("Server Started....");

		}catch(Exception e){
			System.out.println("Exception Occurred At Server!" + e.getMessage());
		}
	}


}
```

**ServerImpl.java**

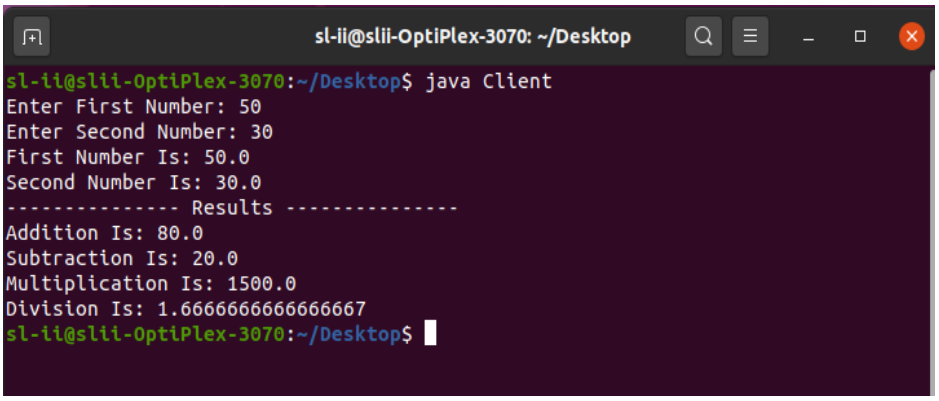
```
import java.rmi.*;
import java.rmi.server.*;

public class ServerImpl extends UnicastRemoteObject
	implements ServerIntf {

		public ServerImpl() throws RemoteException{

		}

		public double Addition(double num1, double num2) throws RemoteException{
			return num1 + num2;
		}


		public double Subtraction(double num1, double num2) throws RemoteException{
			return num1 - num2;
		}


		public double Multiplication(double num1, double num2) throws RemoteException{
			return num1 * num2;
		}


		public double Division(double num1, double num2) throws RemoteException{
			if(num2 != 0){
				return num1/num2;
			}
			else{
				System.out.println("Cannot Divide A Number By Zero!");
			}
			return num1/num2;
		}

	}
```

**SeverIntf.java**

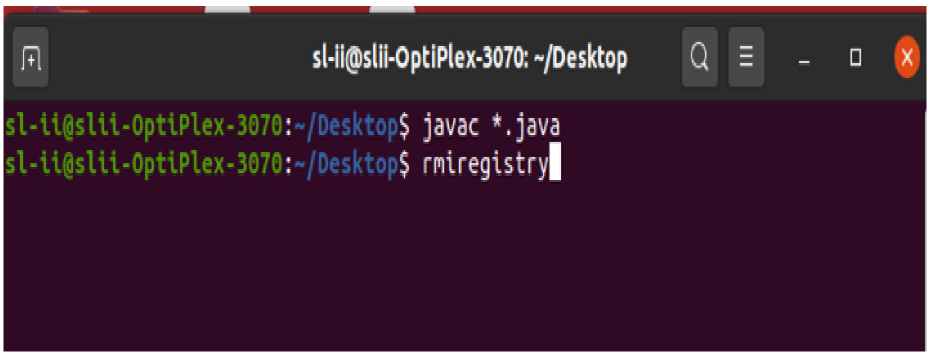
```
        import java.rmi.*;


interface ServerIntf extends Remote{
        // Syntax for method declaration: access_specifier return_type
method_name(arguments...){ return value}

        public double Addition(double num1, double num2) throws RemoteException;
        public double Subtraction(double num1, double num2) throws RemoteException;
        public double Multiplication(double num1, double num2) throws RemoteException;
        public double Division(double num1, double num2) throws RemoteException;
}
```



```
sl-ii@slii-OptiPlex-3070:~/Desktop$ java Server
Server Started....
```



```
sl-ii@slii-OptiPlex-3070:~/Desktop$ java Client
Enter First Number: 50
Enter Second Number: 30
First Number Is: 50.0
Second Number Is: 30.0
--------------- Results ---------------
Addition Is: 80.0
Subtraction Is: 20.0
Multiplication Is: 1500.0
Division Is: 1.6666666666666667
sl-ii@slii-OptiPlex-3070:~/Desktop$
```



```
sl-ii@slii-OptiPlex-3070:~/Desktop$ javac *.java
sl-ii@slii-OptiPlex-3070:~/Desktop$ rmiregistry
```