

ASSIGNMENT NO-1

Client.java

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

public class Client {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            String url = "rmi://localhost/Server";
            ServerIntf serverIntf = (ServerIntf) Naming.lookup(url);

            System.out.println("Enter num1 ");
            int a = sc.nextInt();

            System.out.println("Enter num2 ");
            int b = sc.nextInt();

            // System.out.println("Enter str1 ");
            // String str1 = sc.nextLine();

            // System.out.println("Enter str2 ");
            // String str2 = sc.nextLine();

            System.out.println("Add is " + serverIntf.addition(a, b));
            System.out.println("sub is " + serverIntf.substraction(a, b));
            System.out.println("mul is " + serverIntf.multiplication(a, b));
            System.out.println("div is " + serverIntf.division(a, b));
            System.out.println("square is " + serverIntf.square(a));
            System.out.println("squar root is " + serverIntf.squareroot(b));
```

```

        // System.out.println("Palindrome of string is ");
        // serverIntf.palindrome(str1);

        // System.out.println("String is equal or not");
        // serverIntf.isequalstring(str1, str2);

    } catch (Exception e) {
        System.out.println("Exception at Client " + e);
    }
    sc.close();
}
}

```

Server.java

```

import java.rmi.*;

public class Server {

    public static void main(String[] args) {

        try {
            // Create an instance of ServerImpl class
            ServerImpl serverimpl = new ServerImpl();
            // Bind the instance to the RMI registry with the name "Server"
            Naming.rebind("Server", serverimpl);
            System.out.println(" server started !");

        } catch (Exception e) {
            // Handle any exceptions occurred during the execution
            System.out.println("Eception 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 int addition(int a, int b) throws RemoteException {  
        return a + b;  
    }  
}
```

```
    public int subtraction(int a, int b) throws RemoteException {  
        return a - b;  
    }  
}
```

```
    public int division(int a, int b) throws RemoteException {  
        return a / b;  
    }  
}
```

```
    public int multiplication(int a, int b) throws RemoteException {  
        return a * b;  
    }  
}
```

```
    public int square(int a) throws RemoteException {
```

```

        return a * a;
    }

    public int squareroot(int a) throws RemoteException {
        return (int) (Math.sqrt(a));
    }

    public void palindrome(String str) throws RemoteException {
        StringBuilder sb = new StringBuilder(str);
        sb.reverse();

        if (str.equals(sb.toString()))
            System.out.println("String is Palindrome!");
        else
            System.out.println("String is Not Palindrome!");
    }

    public void isequalstring(String str1, String str2) throws RemoteException {
        if (str1.equals(str2))
            System.out.println("String is equal!");
        else
            System.out.println("String is not equal!");
    }
}

```

ServerIntf.java

```

import java.rmi.*;

interface ServerIntf extends Remote {

```

```
public int addition(int a, int b) throws RemoteException;
```

```
public int subtraction(int a, int b) throws RemoteException;
```

```
public int division(int a, int b) throws RemoteException;
```

```
public int multiplication(int a, int b) throws RemoteException;
```

```
public int square(int a) throws RemoteException;
```

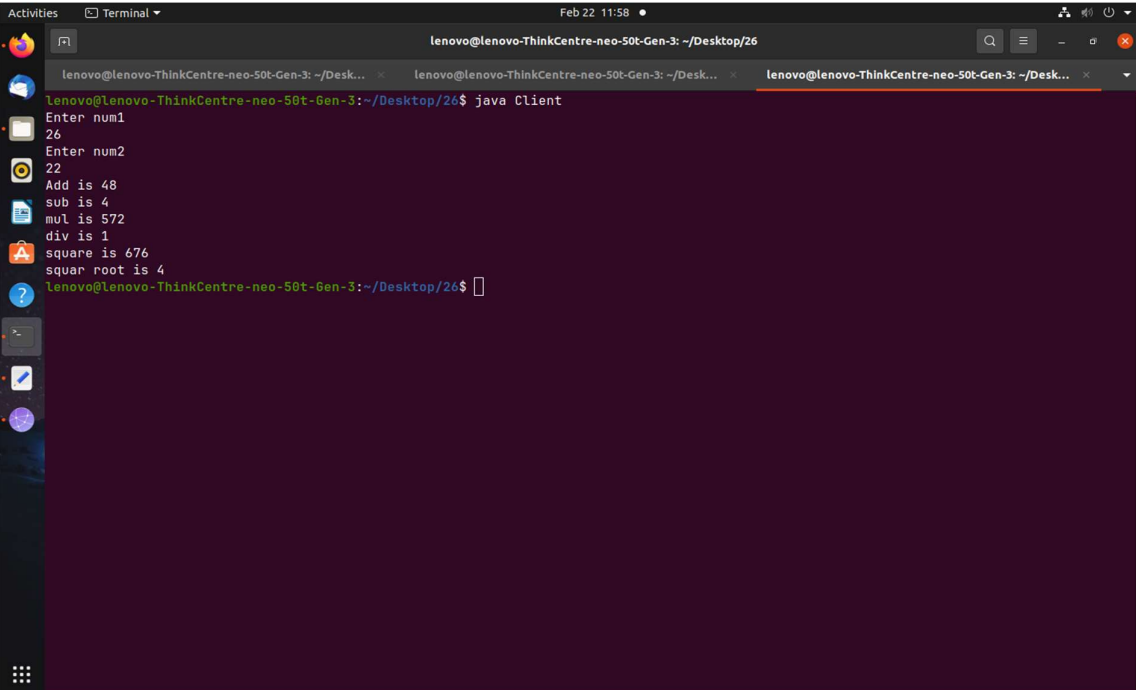
```
public int squareroot(int a) throws RemoteException;
```

```
public void palindrome(String str) throws RemoteException;
```

```
public void isequalstring(String str1, String str2) throws RemoteException;
```

```
}
```

Output



```
lenovo@lenovo-ThinkCentre-neo-50t-Gen-3: ~/Desktop/26
lenovo@lenovo-ThinkCentre-neo-50t-Gen-3:~/Desktop/26$ java Client
Enter num1
26
Enter num2
22
Add is 48
sub is 4
mul is 572
div is 1
square is 676
squar root is 4
lenovo@lenovo-ThinkCentre-neo-50t-Gen-3:~/Desktop/26$
```