

## Code:

### 1. AddServer.java

```
J AddServerIntf.java J AddServerImpl.java J AddServer.java X J AddClient.java
J AddServer.java > AddServer > main(String[])
1  import java.rmi.*;
2
3  public abstract class AddServer {
4      Run | Debug
5      public static void main(String[] args) {
6          try{
7              AddServerImpl addServerImpl = new AddServerImpl();
8              Naming.bind("AddServer", addServerImpl);
9          } catch (Exception e) {
10             System.out.println("Exception " + e);
11         }
12     }
13 }
```

### 2. AddClient.java

```
J AddServerIntf.java J AddServerImpl.java J AddServer.java J AddClient.java X
J AddClient.java > ...
1  import java.rmi.*;
2
3  public class AddClient {
4      Run | Debug
5      public static void main(String[] args) {
6          try{
7              String addServerURL = "rmi://" + args[0] + "/AddServer";
8              AddServerIntf addServerIntf = (AddServerIntf)Naming.lookup(addServerURL);
9              System.out.println("The first number is: " + args[1]);
10             double d1 = Double.valueOf(args[1]).doubleValue();
11             System.out.println("The second number is: " + args[2]);
12             double d2 = Double.valueOf(args[2]).doubleValue();
13             System.out.println("The sum is: " + addServerIntf.add(d1, d2));
14         } catch (Exception e) {
15             System.out.println("Exception " + e);
16         }
17     }
18 }
19 }
20 }
```

### 3. AddServerIntf.java

```
J AddServerIntf.java X J AddServerImpl.java J AddServer.java J AddClient.java
J AddServerIntf.java AddServerIntf > add(double, double)
1 import java.rmi.*;
2
3 public interface AddServerIntf extends Remote {
4     double add(double d1, double d2) throws RemoteException;
5 }
```

### 4. AddServerImpl.java

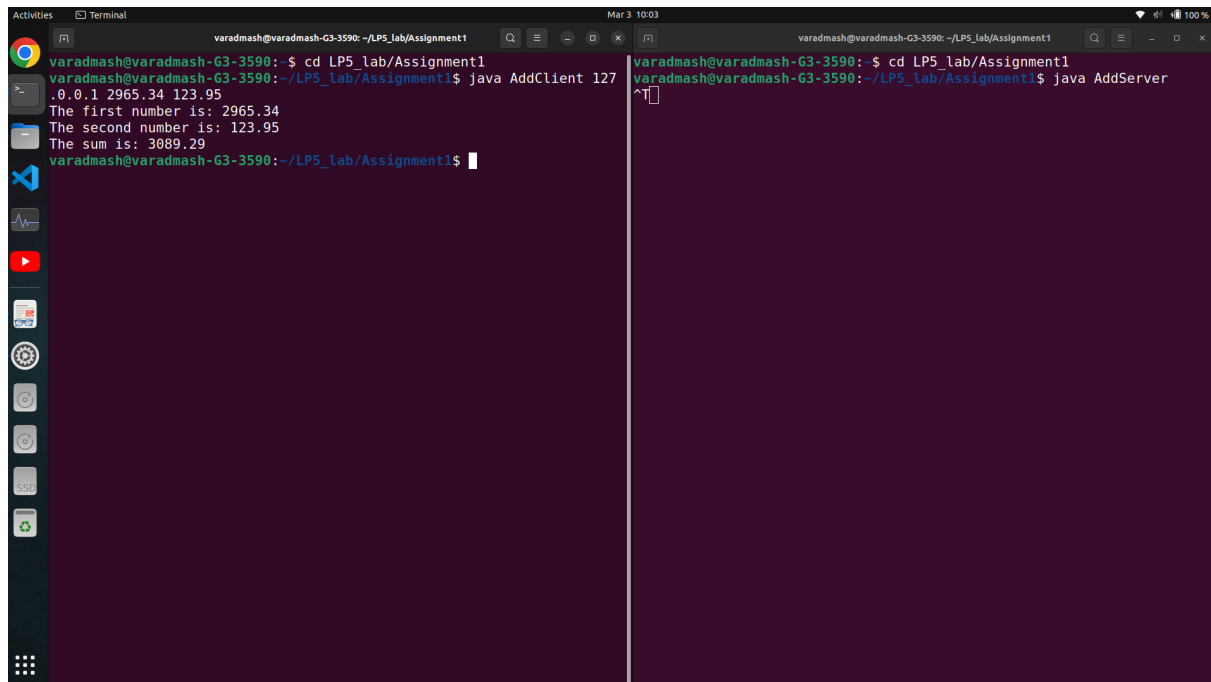
```
J AddServerIntf.java J AddServerImpl.java X J AddServer.java J AddClient.java
J AddServerImpl.java > ...
1 import java.rmi.*;
2 import java.rmi.server.*;
3
4 public class AddServerImpl extends UnicastRemoteObject implements AddServerIntf {
5
6     public AddServerImpl() throws RemoteException {}
7
8     @Override
9     public double add(double d1, double d2) throws RemoteException {
10         return (d1 + d2);
11     }
12
13 }
14
```

## Output:

### 1. Compilation, registration and rmi registry server start

```
varadmash@varadmash-G3-3590: ~/LP5_lab/Assignment1
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment1$ javac *.java
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment1$ rmic AddServerImpl
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment1$ rmiregistry
```

## 2. Server start and client invocation



The image shows two terminal windows side-by-side. The left window shows the execution of the `AddClient` class, which prints the first number (2965.34), the second number (123.95), and their sum (3089.29). The right window shows the execution of the `AddServer` class, which is currently waiting for input, indicated by a cursor and a carriage return character (^).

```
varadmash@varadmash-G3-3590:~$ cd LP5_lab/Assignment1
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment1$ java AddClient 127
.0.0.1 2965.34 123.95
The first number is: 2965.34
The second number is: 123.95
The sum is: 3089.29
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment1$

varadmash@varadmash-G3-3590:~$ cd LP5_lab/Assignment1
varadmash@varadmash-G3-3590:~/LP5_lab/Assignment1$ java AddServer
^T
```