Experiment No: 3

code:

Step 1: Create the Interface

nano AddI.java

```
import java.rmi.*;
public interface AddI extends Remote {
   public int add(int a, int b) throws RemoteException;
}
```

step 2:Create the Server Implementation

nano AddServer.java

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

public class AddServer extends UnicastRemoteObject implements AddI {
   public AddServer() throws RemoteException {}

   public int add(int a, int b) {
      return a + b;
   }
}
```

Step 3 : Create the Registry and Bind the Server

nano RegisterMe.java

```
import java.rmi.*;
public class RegisterMe {
   public static void main(String args[]) {
      try {
         AddServer obj = new AddServer();  // create server object
         Naming.rebind("add", obj);       // bind the object to the name "add"
         System.out.println("Registered successfully");
      } catch (Exception e) {
         System.out.println(e);
      }
   }
}
```

Step 4 : Create the Client Program

```
nano AddClient.java
```

```
import java.rmi.*;
public class AddClient {
   public static void main(String args[]) {
      try {
         int a = 10, b = 20;
         AddI AddIObj = (AddI) Naming.lookup("rmi://localhost/add"); // lookup the server object
         int res = AddIObj.add(a, b);  // call the remote method
         System.out.println("Addition of two numbers: " + res);
      } catch (Exception e) {
         System.out.println(e);
      }
   }
}
```

output:

base) mml@mml-Vostro-3470:~/rmi_example$ java AddClient

Addition of two numbers: 30