**1 RMI ECHO Program**

**Echo Program : -**

**EchoInterface :-**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface EchoInterface extends Remote{ String echo(String s) throws RemoteException;

}

**EchoImpl : -**

import java.rmi.Remote;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

public class EchoImpl extends UnicastRemoteObject implements EchoInterface{

public EchoImpl() throws RemoteException{

}

public String echo(String s1) {

System.out.println("REMOTE SERVICE: Remote Client Request Message is : "+s1);

return s1;

} }

**EchoServer : -**

import java.rmi.Remote;

import java.rmi.RemoteException; import java.rmi.Naming;

import java.net.MalformedURLException; import java.rmi.registry.LocateRegistry;

public class EchoServer {

public EchoServer() throws RemoteException {

}

public static void main(String args[]) throws RemoteException {

EchoImpl echoObj=new EchoImpl(); int port=Integer.parseInt(args[1]);

try {

LocateRegistry.createRegistry(port); System.out.println("\n RMI registry created \n");

String host=args[0];

String bindLocation="//"+host+":"+port+"/"+args[2]; Naming.bind(bindLocation,echoObj); System.out.println("\nRMI server ready at "+bindLocation);

**}**

catch(Exception e) {

e.printStackTrace();

} } }

**EchoClient :-**

import java.rmi.\*; import java.io.\*;

import java.net.MalformedURLException;

public class EchoClient {

public static void main(String args[]) {

String connectLocation="//"+args[0]+":"+Integer.parseInt(args[1])+"/"+args[2];

EchoInterface eintf=null;

try {

System.out.println("\n Connecting the client at: "+connectLocation);

eintf=(EchoInterface)Naming.lookup(connectLocation); BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("\nCLI: Enter the request message to send to remote service:");

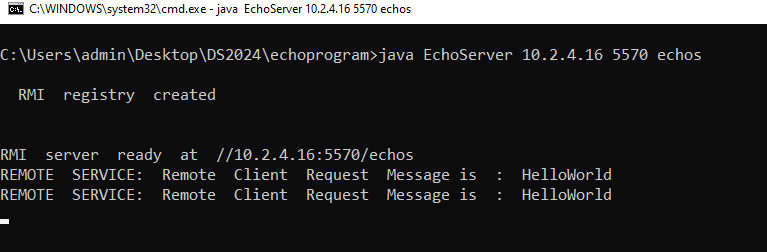
String s=br.readLine(); String response=eintf.echo(s);

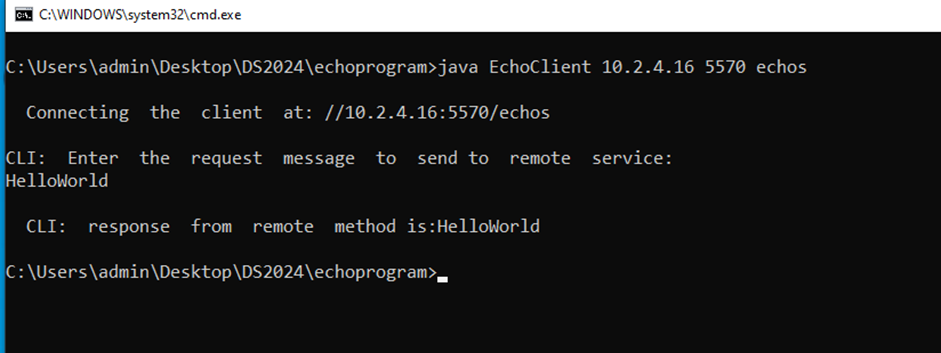
System.out.println("\n CLI: response from remote method is:"+response);

}

catch(Exception e) {

e.printStackTrace(); } } }



****

**2 RMI Program to Reverse a String**

**ReverseInterface:-**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface ReverseInterface extends Remote {

String reverseStr(String s) throws RemoteException;

}

**ReverseImpl :-**

import java.rmi.Remote;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

public class ReverseImpl extends UnicastRemoteObject implements ReverseInterface {

public ReverseImpl() throws RemoteException {

}

public String reverseStr(String s1) {

System.out.println("REMOTE SERVICE : Remote Client Request Message IS : "+s1);

StringBuilder sb=new StringBuilder(s1);

String response = sb.reverse().toString();

return response;

} }

**ReverseServer : -**

import java.rmi.Remote;

import java.rmi.RemoteException; import java.rmi.Naming;

import java.net.MalformedURLException; import java.rmi.registry.LocateRegistry;

public class ReverseServer {

public ReverseServer() throws RemoteException {

}

public static void main(String args[]) throws RemoteException {

ReverseImpl reverseObj=new ReverseImpl(); int port=Integer.parseInt(args[1]);

try {

LocateRegistry.createRegistry(port); System.out.println("\n RMI registry created \n");

String host=args[0];

String bindLocation="//"+host+":"+port+"/"+args[2]; Naming.bind(bindLocation,reverseObj); System.out.println("\nRMI server ready at "+bindLocation);

}

catch(Exception e) {

e.printStackTrace(); }

} }

**ReverseClient :-**

import java.rmi.\*; import java.io.\*;

import java.net.MalformedURLException;

public class ReverseClient {

public static void main(String args[]) {

String connectLocation="//"+args[0]+":"+Integer.parseInt(args[1])+"/"+args[2];

ReverseInterface revintf=null;

try {

System.out.println("\n Connecting the client at: "+connectLocation);

revintf=(ReverseInterface)Naming.lookup(connectLocation);

BufferedReader br=new BufferedReader(new

InputStreamReader(System.in));

System.out.println("\nCLI: Enter the request message to send to remote service:");

String s=br.readLine();

String response=revintf.reverseStr(s);

System.out.println("\n CLI: response from remote method is:"+response);

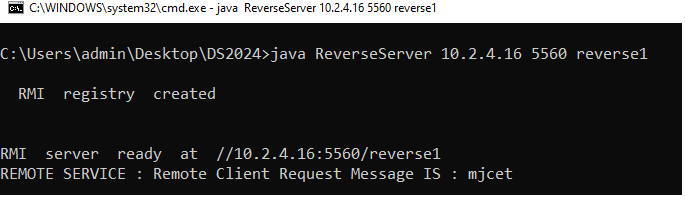
}

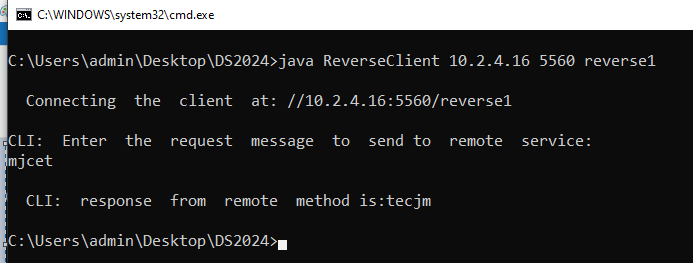
catch(Exception e) {

e.printStackTrace(); }

} }

OP:-





**3 Concatenatiing Two Strings**

**ConcatInterface :-**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface ConcatInterface extends Remote{

String concatenate(String s1,String s2) throws RemoteException;

}

**ConcatImpl :-**

import java.rmi.Remote;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

public class ConcatImpl extends UnicastRemoteObject implements ConcatInterface{

public ConcatImpl() throws RemoteException{

}

public String concatenate(String s1,String s2)

{

System.out.println("REMOTE SERVICE: Remote Client Request Message is : "+s1+" and "+s2);

return s1+s2;

} }

**ConcatServer :-**

import java.rmi.Remote;

import java.rmi.RemoteException; import java.rmi.Naming;

import java.net.MalformedURLException; import java.rmi.registry.LocateRegistry;

public class ConcatServer{

public ConcatServer() throws RemoteException{

}

public static void main(String args[]) throws RemoteException

{

ConcatImpl concatObj=new ConcatImpl(); int port=Integer.parseInt(args[1]); try{

LocateRegistry.createRegistry(port); System.out.println("\n RMI registry created \n"); String host=args[0];

String bindLocation="//"+host+":"+port+"/"+args[2]; Naming.bind(bindLocation,concatObj); System.out.println("\nRMI server ready at "+bindLocation);

}

catch(Exception e)

{

e.printStackTrace();

} } }

**ConcatClient :-**

import java.rmi.\*; import java.io.\*;

import java.net.MalformedURLException;

public class ConcatClient {

public static void main(String args[])

{

String connectLocation="//"+args[0]+":"+Integer.parseInt(args[1])+"/"+args[2];

ConcatInterface cintf=null; try{

System.out.println("\n Connecting the client at: "+connectLocation);

cintf=(ConcatInterface)Naming.lookup(connectLocation);

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("\nCLI: Enter two strings to send to remote service:");

String s1=br.readLine(); String s2=br.readLine();

String response=cintf.concatenate(s1,s2);

System.out.println("\n CLI: response from remote method is:"+response);

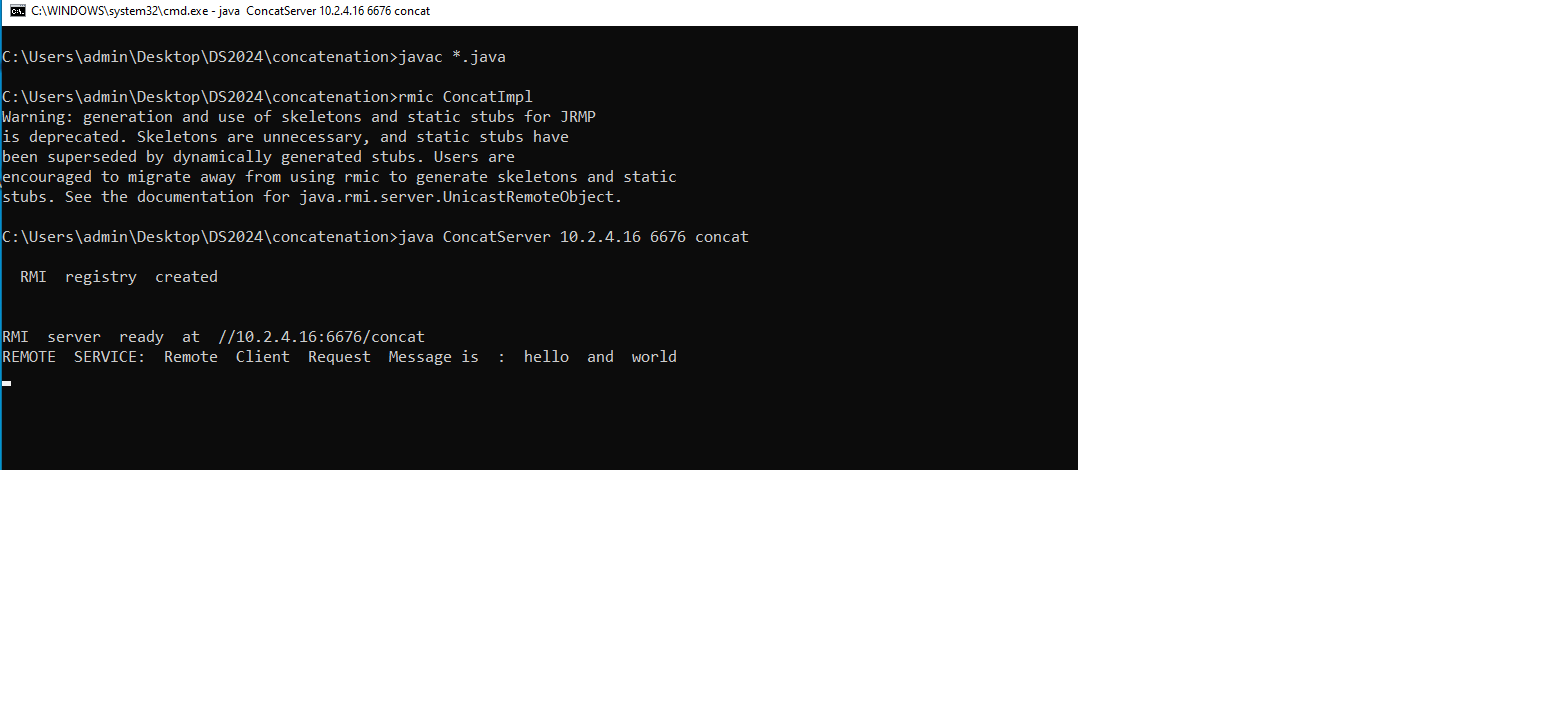
}

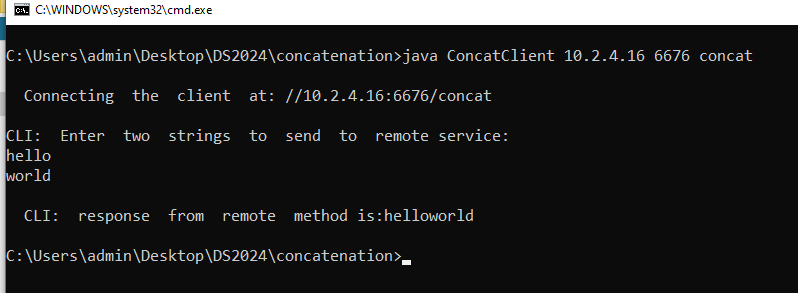
catch(Exception e)

{

e.printStackTrace();

} } }





**4 RMI Program to find Sum of Two Numbers:-**

**AddInterface.java:-**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface AddInterface extends Remote{ int add(int a,int b) throws RemoteException;

}

**AddImpl.java**

import java.rmi.Remote;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

public class AddImpl extends UnicastRemoteObject implements AddInterface{ public AddImpl() throws RemoteException{

}

public int add(int a,int b) {

System.out.println("REMOTE SERVICE: Remote Client Request Message is : "+a+" and "+b);

return a+b;

} }

**AddServer.java**

import java.rmi.Remote;

import java.rmi.RemoteException;

import java.rmi.Naming;

import java.net.MalformedURLException;

import java.rmi.registry.LocateRegistry;

public class AddServer{

public AddServer() throws RemoteException{ }

public static void main(String args[]) throws RemoteException {

AddImpl addObj=new AddImpl();

int port=Integer.parseInt(args[1]);

try {

LocateRegistry.createRegistry(port); System.out.println("\n RMI registry created \n"); String host=args[0];

String bindLocation="//"+host+":"+port+"/"+args[2]; Naming.bind(bindLocation,addObj); System.out.println("\nRMI server ready at "+bindLocation);

}

catch(Exception e) {

e.printStackTrace();

} } }

**AddClient.java :-**

import java.rmi.\*; import java.io.\*;

import java.net.MalformedURLException;

public class AddClient {

public static void main(String args[]) {

String connectLocation="//"+args[0]+":"+Integer.parseInt(args[1])+"/"+args[2];

AddInterface aintf=null; try{

System.out.println("\n Connecting the client at: "+connectLocation);

aintf=(AddInterface)Naming.lookup(connectLocation);

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("\nCLI: Enter two numbers to send to remote service:");

int a=Integer.parseInt(br.readLine().trim()); int b=Integer.parseInt(br.readLine().trim());

int response=aintf.add(a,b);

System.out.println("\n CLI: response from remote method is(sum):"+response);

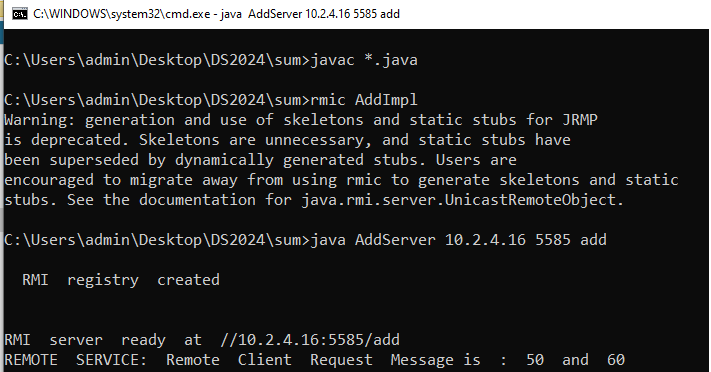
}

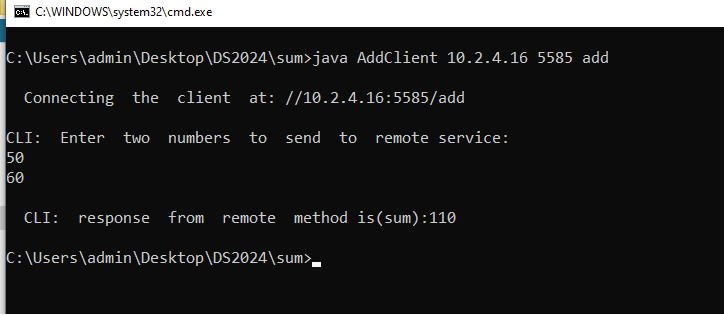
catch(Exception e)

{

e.printStackTrace();

} } }





**5.Implement a RMI program to find length of a given String.**

**LengthInterface.java**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface LengthInterface extends Remote{ int strLength(String s) throws RemoteException;

}

**LengthImpl.java :-**

import java.rmi.Remote;

import java.rmi.RemoteException;

import java.rmi.server.UnicastRemoteObject;

public class LengthImpl extends UnicastRemoteObject implements LengthInterface{

public LengthImpl() throws RemoteException{

}

public int strLength(String s1) {

System.out.println("REMOTE SERVICE: Remote Client Request Message is : "+s1);

return s1.length();

} }

**LengthServer.java :-**

import java.rmi.Remote;

import java.rmi.RemoteException; import java.rmi.Naming;

import java.net.MalformedURLException; import java.rmi.registry.LocateRegistry;

public class LengthServer{

public LengthServer() throws RemoteException { }

public static void main(String args[]) throws RemoteException {

LengthImpl lengthObj=new LengthImpl(); int port=Integer.parseInt(args[1]);

try {

LocateRegistry.createRegistry(port); System.out.println("\n RMI registry created \n");

String host=args[0];

String bindLocation="//"+host+":"+port+"/"+args[2]; Naming.bind(bindLocation,lengthObj); System.out.println("\nRMI server ready at "+bindLocation); }

catch(Exception e) {

e.printStackTrace();

} } }

**LengthClient.java:-**

import java.rmi.\*; import java.io.\*;

import java.net.MalformedURLException;

public class LengthClient {

public static void main(String args[])

{

String connectLocation="//"+args[0]+":"+Integer.parseInt(args[1])+"/"+args[2];

LengthInterface lintf=null;

try {

System.out.println("\n Connecting the client at: "+connectLocation);

lintf=(LengthInterface)Naming.lookup(connectLocation); BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.println("\nCLI: Enter the request message to send to remote service:");

String s=br.readLine();

int response=lintf.strLength(s);

System.out.println("\n CLI: response from remote method is:"+response);

}

catch(Exception e) {

e.printStackTrace();

} } }

