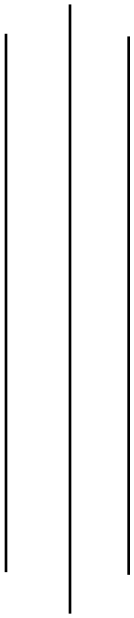

Programme Name: **BCS HONS**

Course Code: **CSC 2624**

Course Name: **Distributed And Parallel Computing**

Internal Examination

Date of Submission: **8/26/2021**

Submitted By:

Student Name**: Dipesh Tha Shrestha**

IUKL ID: **041902900028**

Semester**: Fourth Semester**

Intake**: September 2019**

Submitted To:

Faculty Name**: Manoj Gautam**

Department**: LMS**

1. **Write a java program with a two class Server.java and Client.java to implement a network programming. Server must listen on the port 8000 and should be able to accept client request and send back the original data to the client.**

Answer:

As we know, we need 2 folders (client and server) to run the above program. In given folder it will have its own files which will help them to connect with each other. Therefore, Code to to accept client request and send back the original data to the client is given below:

Server.java

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {
public static void main(String[] args) throws IOException { String clientSentence
;
String capitalizedSentence;
ServerSocket welcomeSocket = new ServerSocket(8000); System.out.println("Server i
s
listening at port: " + 8000); while (true) {
Socket connectionSocket = welcomeSocket.accept();
BufferedReader fromClient = new BufferedReader(new InputStreamReader(connecti
onSocket.getInputStream()));
clientSentence = fromClient.readLine(); capitalizedSentence =
clientSentence.toUpperCase() + "\n";
DataOutputStream toClient = new DataOutputStream(connectionSocket.getOutput
Stream());
toClient.writeBytes(capitalizedSentence);
}
}

}
```

Client.java

```
import java.io.*;
import java.net.*;

public class Client {
public static void main(String[] args) throws Exception { try {
String sentence;
String modifiedSentence;
BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.
in));
Socket clientSocket = new Socket("127.0.0.1", 8000);
DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputSt
ream());
sentence = inFromUser.readLine(); outToServer.writeBytes(sentence + "\n");
InputStreamReader ins = new InputStreamReader(clientSocket.getInputStream());
BufferedReader inFromServer = new BufferedReader(ins);
modifiedSentence = inFromServer.readLine();
System.out.println("The data received from the server is " + modifiedSentence);
outToServer.flush();
outToServer.close(); clientSocket.close();
} catch (Exception e) { System.out.println(e);
}
}
}
```

1. **Write a JAVA RMI Server and Client program to calculate Power of a number such that Rmiregistry listen on port 9200**.

Answer:

As we know, we need 2 folders (client and server) to run the above program. In given folder it will have its own files which will help them to connect with each other. Folder client will have 3 files named as Client.java, Number.java and RemoteCalcObject.java whereas Folder server will have Server.java, Number.java and NumberImpl.java. Therefore, Code for above program is given below:

Folder Server

Server.java

```java
import java.rmi.server.UnicastRemoteObject;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class Server {
    public static void main(String[] args) {
        try {
            NumberImpl n1 = new NumberImpl(4);
            Number stub1 = (Number) UnicastRemoteObject.exportObject(n1, 0);
            Registry registry = LocateRegistry.getRegistry("127.0.0.1", 9300);
            registry.bind("number", stub1);
        } catch (Exception e) {
            System.out.println("Error :" + e);
        }

    }
}
```

Number.java

```java
import java.rmi.*;

public interface Number extends Remote {
    public double getNum() throws RemoteException;
}
```

NumberImpl.java

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

public class NumberImpl implements Number{

    double numm;

    NumberImpl(double newnumm) throws RemoteException{
        this.numm = newnumm;
    }
    public double getNum() throws RemoteException{
        return this.numm;
    }
```

```
}
```

Folder Client

Client.java

```
import java.rmi.*;
import java.rmi.registry.*;

public class Client {
    public static void main(String[] args) throws RemoteException, NotBoundExcept
ion {
     try {
        Registry remoteRegistry = LocateRegistry.getRegistry("127.0.0.1", 9200);
         Number numm = (Number) remoteRegistry.lookup("number");
         RemoteCalcObject remoteCalcObject = new RemoteCalcObject();
         double finalnum = remoteCalcObject.computerPower(numm.getNum());
          System.out.println("The power of " + numm.getNum() + " by 2 is " + fina
lnum);
        } catch (Exception e) {
            System.out.println("Clinet error occoured " + e.toString());
        }

    }
}
```

Number.java

```
import java.rmi.*;

public interface Number extends Remote {
    public double getNum() throws RemoteException;
}
```

RemoteCalcObject.java

```
import java.lang.Math;

class RemoteCalcObject {
    RemoteCalcObject() {
```

```
    }

    public double computerPower(double num) {
        return Math.pow(num, 2);
    }
}
```

Output:

```
C:\Users\dell\Desktop\internal_exam\Server>javac *.java

C:\Users\dell\Desktop\internal_exam\Server>start rmiregistry 9200

C:\Users\dell\Desktop\internal_exam\Server>java Server
```

```
C:\Users\dell\Desktop\internal_exam\Client>javac *.java

C:\Users\dell\Desktop\internal_exam\Client>java Client
The power of 77.0 by 2 is 5929.0

C:\Users\dell\Desktop\internal_exam\Client>
```

Thank You