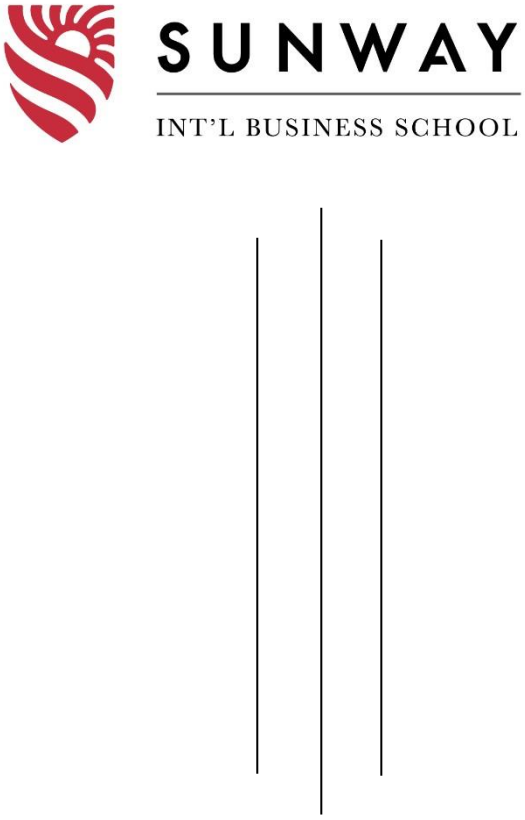

Programme Name: BCS(Hons.)

Course Name: Distributed and parallel computing

First Examination

Date of Submission: 08/26/2021

Submitted By:

Student Name: Ishup Khadka

Semester: 4th

Intake: September, 2019

Submitted To:

Faculty Name: Prakash Chandra

Department: LMS

Question number 1

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);

}

}

}
```

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 is
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);

}

}


}
```

Question number 2

For Client Side

Client.java

```
import java.rmi.registry.LocateRegistry; import java.rmi.registry.Registry; import
java.util.Scanner;


public class Client {
```

```
public static void main(String[] args) {


try {

Scanner sc = new Scanner(System.in); System.out.println("Enter the number: "); double
number = sc.nextDouble(); System.out.print("Enter the power to raise: "); int
powerToRaise = sc.nextInt();


Registry registry = LocateRegistry.getRegistry("127.0.0.1", 9200);

Product RemoteCalcObject = (Product) registry.lookup("computerPower");


double powerobj = RemoteCalcObject.computerPower(number, powerToRaise);

System.out.println("The final value of " + number + " to the power of " + powerToRa ise
+ " = " + powerobj);

sc.close();


} catch (Exception e) {

System.out.println("Client Exception : " + e.toString()); e.printStackTrace();

}

}


}
```

For Client Side and Server Side:

Product.java

```
import java.rmi.Remote;

import java.rmi.RemoteException;


public interface Product extends Remote {
```

```
public double computerPower(double number, int powerToRaise) throws RemoteExcep
tion;

}

import java.rmi.RemoteException;


public class ProductImpl implements Product { String lookupstr;


public ProductImpl(String lookuptext) throws RemoteException { this.lookupstr =
lookuptext;


}


public static void main(String[] args) {


}


public double computerPower(double number, int powerToRaise) throws RemoteExcep
tion {

return Math.pow(number, powerToRaise);


}


}
```

Server.java

For Server Side:

ProductImpl.java

```
import java.rmi.registry.Registry;

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


public class Server {

public static void main(String[] args) { try {

System.out.println("=========================Server is booting
now=========

==============");


System.setProperty("java.rmi.server.hostname", "127.0.0.1"); ProductImpl obj1 = new
ProductImpl("powermethod");

Product stub = (Product) UnicastRemoteObject.exportObject(obj1, 0); Registry registry
= LocateRegistry.getRegistry("127.0.0.1", 9200); registry.bind("computerPower", stub);

System.err.println("Now server is ready");

} catch (Exception e) {

System.out.println("Some server error occured!!!" + e);

}


}


}
```