RAMNIRANJAN JHUNJHUNWALA COLLEGE

GHATKOPAR (W), MUMBAI - 400 086

DEPARTMENT OF INFORMATION TECHNOLOGY

2020 - 2021

M.Sc.(I.T.) SEM I

Distributed System

Name: SHREYANS UPADHYAY

Roll No.: 18

Hindi Vidya Prachar Samiti's

RAMNIRANJAN

JHUNJHUNWALA COLLEGE

(AUTONOMOUS)

Opposite Ghatkopar Railway Station, Ghatkopar West, Mumbai-400086

CERTIFICATE

This is to certify that Mr. **Upadhyay Shreyans Indresh** with Roll No. **18** has successfully completed the necessary course of experiments in the subject of **Distributed Systems** during the academic year **2020 – 2021** complying with the requirements of **RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE AND COMMERCE**, for the course of **M.Sc. (IT)** semester -I.

Internal Examiner External Examiner

Head of Department College Seal

INDEX:

Practical No 1

Aim: Write a program for implementing Client Server Communication model

Client Server Communication model

Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs, which share their resources with clients. A client does not share any of its resources, but it requests content or service from a server. Clients therefore initiate communication sessions with servers, which await incoming requests. Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.

The client-server characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services. Servers are classified by the services they provide. For example, a web server serves web pages and a file server serves computer files. A shared resource may be any of the server computer's software and electronic components, from programs and data to processors and storage devices. The sharing of resources of a server constitutes a service.

Example 1A: A client server based program using TCP to find if a number entered is prime.

TCPServerPrimeNumber.java

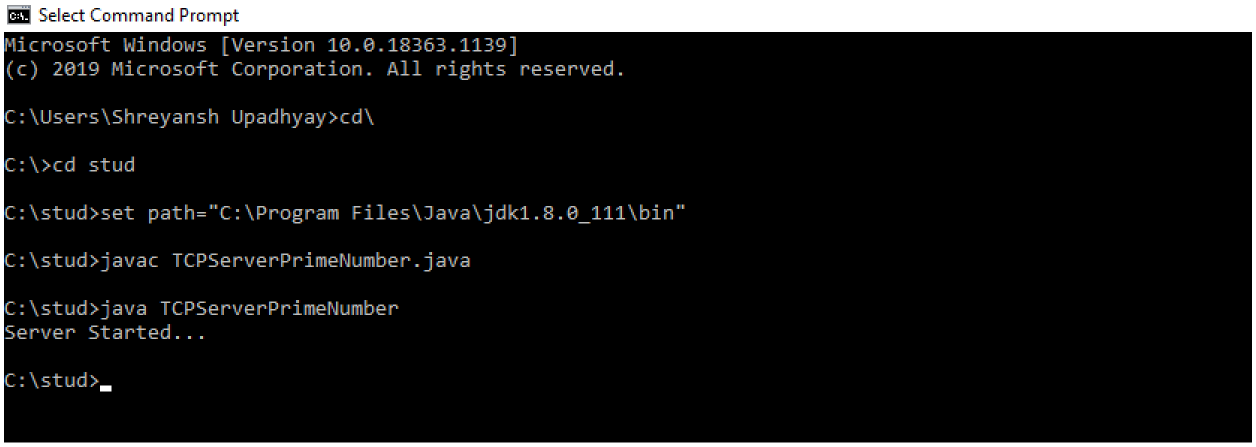
```
import java.net.*;
import java.io.*;
public class TCPServerPrimeNumber
{
	public static void main(String args[])
	{
		try
		{
			ServerSocket ss= new ServerSocket(8001);
			System.out.println("Server Started...");
			Socket s=ss.accept();
			DataInputStream in= new DataInputStream(s.getInputStream());
			int x= in.readInt();
			DataOutputStream otc= new DataOutputStream(s.getOutputStream());
			int y=x/2;
			if(x==1)
			{
				otc.writeUTF(x+ "is neither prime nor composite");
```

```
                                System.exit(0);
                        }
                        for(int i=2;i<=y;i++)
                        {
                                if(x%i!=0)
                                {
                                        otc.writeUTF(x+ "is a prime number.");
                                }
                                else
                                {
                                        otc.writeUTF(x+ "is not a prime number");
                                }
                        }
                }
                catch(Exception e)
                {
                        System.out.println(e.toString());
                }
        }
}
```

TCPClientPrimeNumber.java

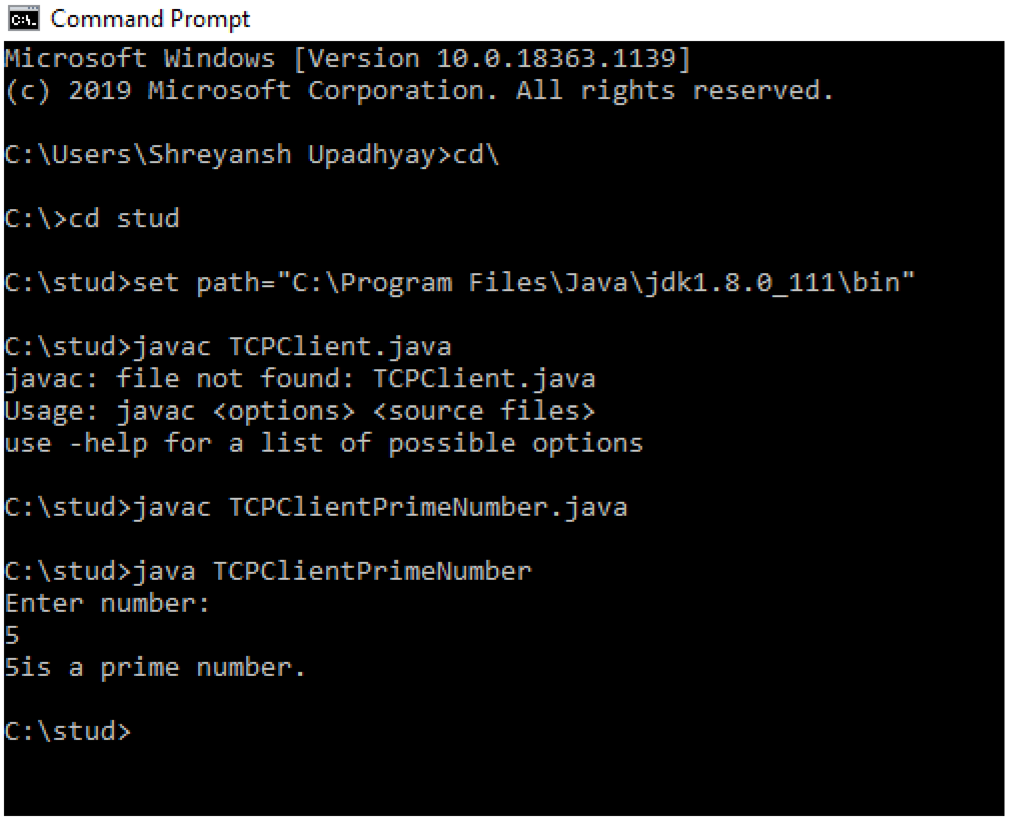
```
import java.net.*;
import java.io.*;
public class TCPClientPrimeNumber
{
        public static void main(String args[])
        {
                try
                {
                        Socket cs= new Socket("LocalHost", 8001);
                        BufferedReader infu=new BufferedReader(new
InputStreamReader(System.in));
                        System.out.println("Enter number:");
                        int a=Integer.parseInt(infu.readLine());
                        DataOutputStream out= new DataOutputStream(cs.getOutputStream());
                        out.writeInt(a);
                        DataInputStream in= new DataInputStream(cs.getInputStream());
                        System.out.println(in.readUTF());
                        cs.close();
                }
                catch(Exception e)
                {
                        System.out.println(e.toString());
                }
        }
}
```

Output



Select Command Prompt

```
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TCPServerPrimeNumber.java

C:\stud>java TCPServerPrimeNumber
Server Started...

C:\stud>
```



Command Prompt

```
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TCPClient.java
javac: file not found: TCPClient.java
Usage: javac <options> <source files>
use -help for a list of possible options

C:\stud>javac TCPClientPrimeNumber.java

C:\stud>java TCPClientPrimeNumber
Enter number:
5
5is a prime number.

C:\stud>
```

Example 1B: A client server TCP based chatting application

TCPServerChat.java

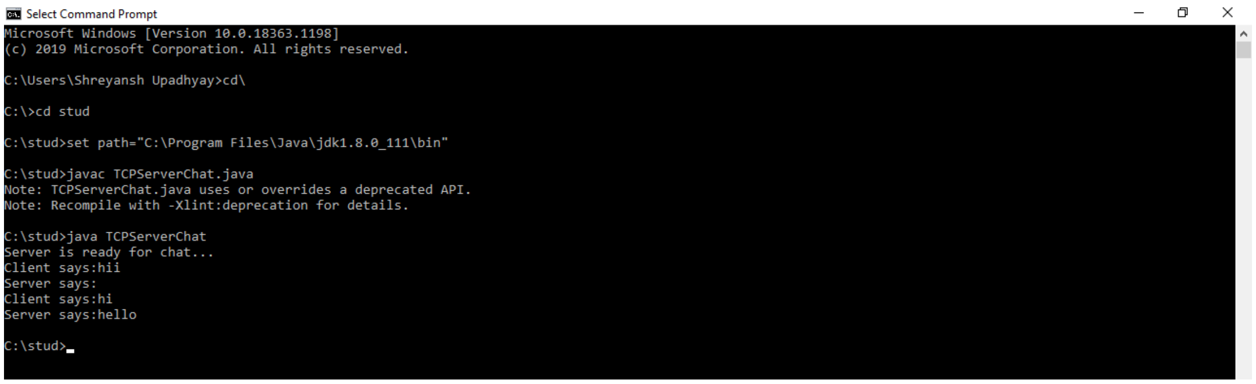
```
import java.net.*;
import java.io.*;
public class TCPServerChat
{
	public static void main(String args[])
	{
		try
		{
			ServerSocket ss= new ServerSocket(8001);
			System.out.println("Server is ready for chat...");
			Socket s= ss.accept();
			BufferedReader infu= new BufferedReader(new
InputStreamReader(System.in));
			DataOutputStream ot=new DataOutputStream(s.getOutputStream());
			DataInputStream in=new DataInputStream(s.getInputStream());
			String receieve;
			String send;
			while((receieve= in.readLine())!=null)
			{
				if(receieve.equals("STOP") || receieve.equals("stop") ||
receieve.equals("Stop"))
				break;
				System.out.println("Client says:" +receieve);
				System.out.print("Server says:");
				send=infu.readLine();
				ot.writeBytes(send+"\n");
			}
			infu.close();
			in.close();
			ot.close();
		}
		catch(Exception e)
		{
			System.out.println(e.toString());
		}
	}
}
```

TCPClientChat.java

```
import java.net.*;
import java.io.*;
public class TCPClientChat
{
	public static void main(String args[])
	{
		try
		{
			Socket cs=new Socket("LocalHost", 8001);
			BufferedReader infu= new BufferedReader(new
InputStreamReader(System.in));
			DataOutputStream ot= new DataOutputStream(cs.getOutputStream());
			DataInputStream in= new DataInputStream(cs.getInputStream());
			String send;

			System.out.println("Type STOP/Stop/stop if u want to stop");
			System.out.println("Client says:");
			while((send = infu.readLine())!=null)
			{
				ot.writeBytes(send+"\n");
				if(send.equals("Stop") || send.equals("stop") ||
send.equals("STOP"))
				break;
				System.out.println("Server says:>>" +in.readLine());
				System.out.print("Client says:>>");

			}
			infu.close();
			in.close();
			ot.close();
			cs.close();

		}
		catch(Exception e)
		{
			System.out.println(e.toString());
		}
```

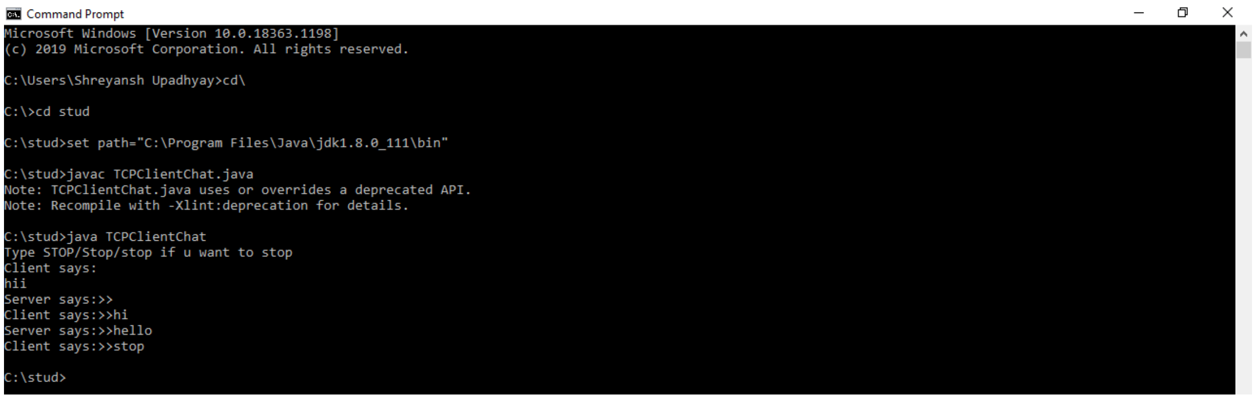
```
        }
}
```

Output



Select Command Prompt

```
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TCPServerChat.java
Note: TCPServerChat.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\stud>java TCPServerChat
Server is ready for chat...
Client says:hii
Server says:
Client says:hi
Server says:hello

C:\stud>
```



Command Prompt

```
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TCPClientChat.java
Note: TCPClientChat.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\stud>java TCPClientChat
Type STOP/Stop/stop if u want to stop
Client says:
hii
Server says:>>
Client says:>>hi
Server says:>>hello
Client says:>>stop

C:\stud>
```

Practical No. 2

Aim: Write a program to show the object communication using RMI

Remote Method Invocation (RMI)

The RMI is an API that provides a mechanism to create distributed application in java. The RMI allows an object to invoke methods on an object running in another JVM. The RMI provides remote communication between the applications using two objects stud and skeleton.
RMI uses stub and skeleton object for communication with the remote object.

stub

The stub is an object, acts as a gateway for the client side. All the outgoing requests are routed through it. It resides at the client side and represents the remote object. When the caller invokes method on the stub object, it does the following tasks:

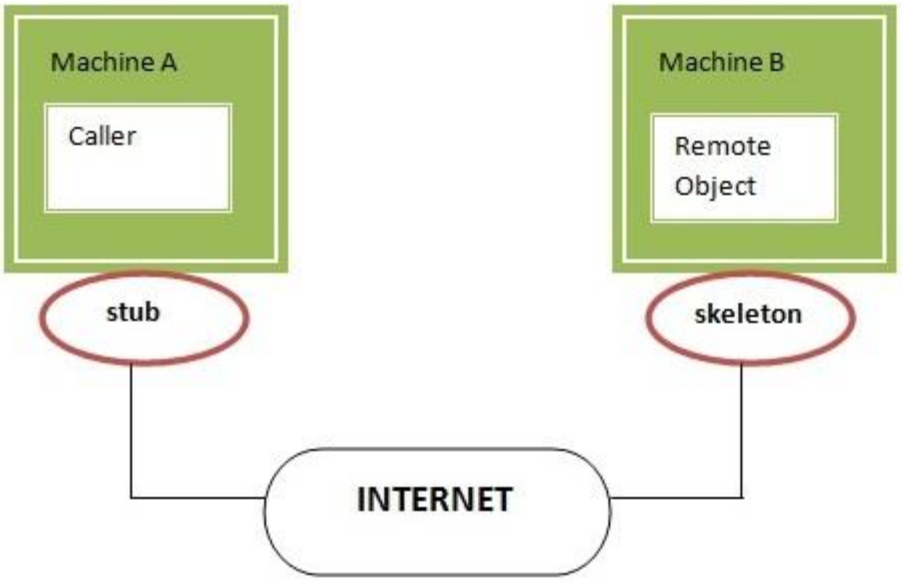
1. It initiates a connection with remote Virtual Machine (JVM),
2. It writes and transmits (marshals) the parameters to the remote Virtual Machine (JVM),
3. It waits for the result
4. It reads (unmarshals) the return value or exception, and
5. It finally, returns the value to the caller.

skeleton

The skeleton is an object, acts as a gateway for the server side object. All the incoming requests are routed through it. When the skeleton receives the incoming request, it does the following tasks:

1. It reads the parameter for the remote method
2. It invokes the method on the actual remote object, and
3. It writes and transmits (marshals) the result to the caller.

In the Java 2 SDK, an stub protocol was introduced that eliminates the need for



skeletons.

Example 2A: A RMI based application program to display current Date and Time

Code

InterDate.java

```
import java.rmi.*;
public interface InterDate extends Remote
{
	public String display() throws RemoteException;
}
```

ServerDate.java

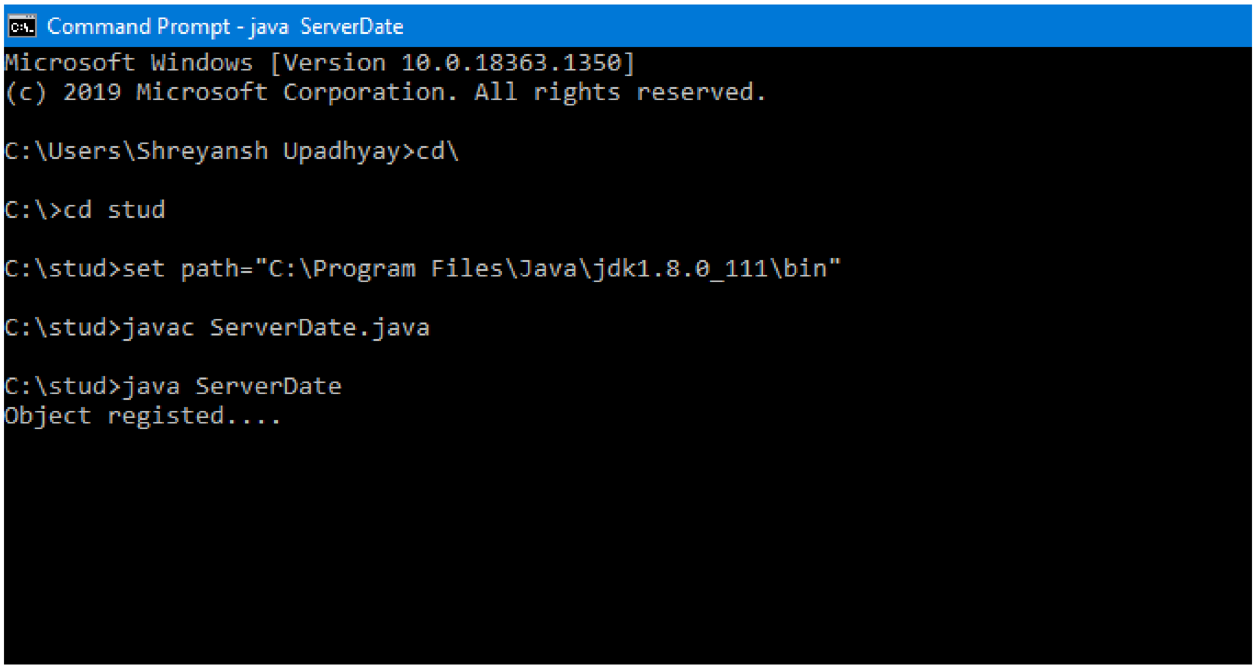
```
import java.rmi.*;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.*;
import java.util.*;
public class ServerDate extends UnicastRemoteObject implements InterDate
{
	public ServerDate() throws RemoteException
	{
	  }
	public String display() throws RemoteException
	{
		String str = "";
		Date d = new Date();
		str = d.toString();
		return str;
	}
	public static void main(String args[]) throws RemoteException
	{
		try
		{
			ServerDate s1= new ServerDate();
			Registry reg = LocateRegistry.createRegistry(5678);
			reg.rebind("DS",s1);
			System.out.println("Object registed....");
		}
		catch(RemoteException e)
		{
			System.out.println("exception"+e);
		}
	}
}
```

ClientDate.java

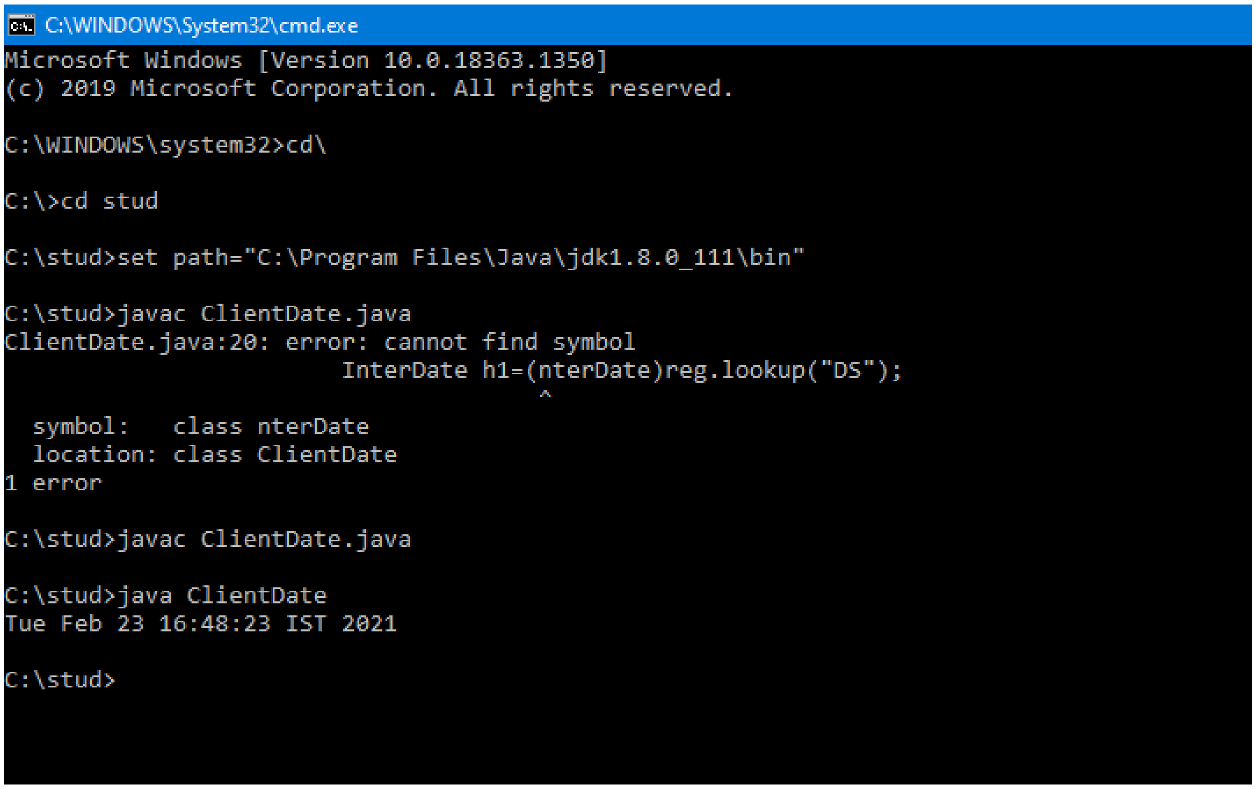
```
import java.rmi.*;
import java.io.*;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.*;

public class ClientDate
{
	public static void main(String args[]) throws RemoteException
	{
		ClientDate c=new ClientDate();
		c.connectRemote();
	}
	private void connectRemote() throws RemoteException
	{
		try
```

```
                    {
                              String s1;
                              Registry reg=LocateRegistry.getRegistry("localhost",5678);
                              InterDate h1=(InterDate)reg.lookup("DS");
                              s1=h1.display();
                              System.out.println(s1);
                    }
                    catch(NotBoundException|RemoteException e)
                    {
                              System.out.println(e);
                    }
          }
}
```

Output



```
Microsoft Windows [Version 10.0.18363.1350]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac ServerDate.java

C:\stud>java ServerDate
Object registed....
```


C:\WINDOWS\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18363.1350]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac ClientDate.java
ClientDate.java:20: error: cannot find symbol
                        InterDate h1=(nterDate)reg.lookup("DS");
                                      ^
  symbol:   class nterDate
  location: class ClientDate
1 error

C:\stud>javac ClientDate.java

C:\stud>java ClientDate
Tue Feb 23 16:48:23 IST 2021

C:\stud>
```

Example 2B: A RMI based application program that converts digits to words, e.g. 123 will be converted to one two three

ServerConverter.java

```
import java.rmi.*;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.*;
public class ServerConverter extends UnicastRemoteObject implements InterfaceConverter
{
        public ServerConverter() throws RemoteException
        {
          }
        public String convertDigit(String no) throws RemoteException
        {
                String str="";
                for(int i=0;i<no.length();i++)
                {
                        int p=no.charAt(i);
                        switch(p)
                        {
                                case 48:
                                        str += "zero ";
                                        break;
                                case 49:
                                        str += "one ";
                                        break;
                                case 50:
                                        str += "two ";
                                        break;
                                case 51:
                                        str += "three ";
                                        break;
                                case 52:
                                        str += "four ";
                                        break;
                                case 53:
                                        str += "five ";
                                        break;
```

```
                    case 54:
                        str += "six ";
                        break;
                    case 55:
                        str += "seven ";
                        break;
                    case 56:
                        str += "eight ";
                        break;
                    case 57:
                        str += "nine ";
                        break;
                }
            }
            return str;
        }
        public static void main(String args[]) throws RemoteException
        {
            try
            {
                ServerConverter s1=new ServerConverter();
                Registry reg=LocateRegistry.createRegistry(5678);
                reg.rebind("Wrd",s1);
                System.out.println("Object Registered...");
            }
            catch(RemoteException e)
            {
                System.out.println("exception" +e);
            }
        }
}
```

ClientConverter.java

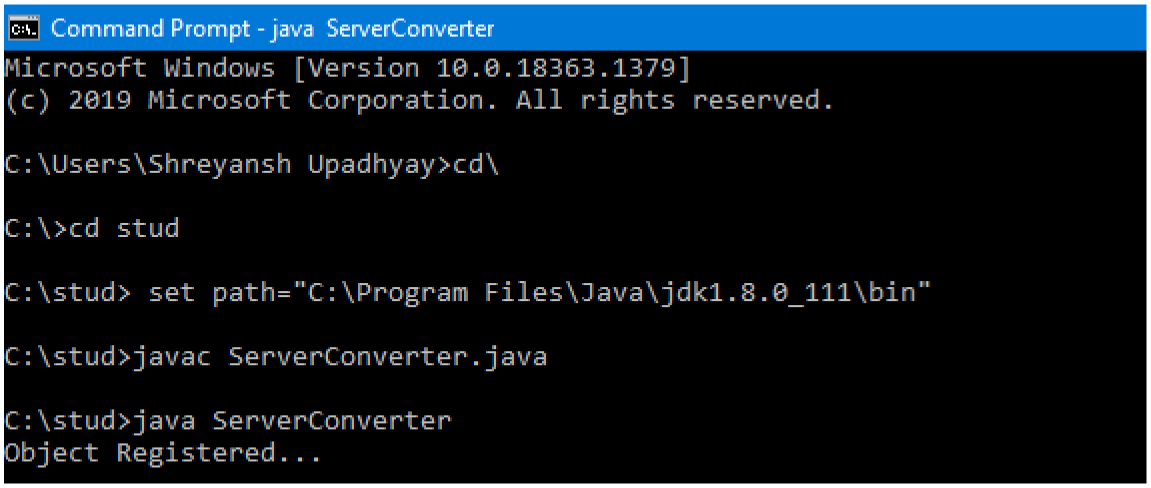
```
import java.rmi.*;
import java.io.*;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class ClientConverter
{
        public static void main(String args[]) throws RemoteException, IOException
```

```
	{
		ClientConverter c=new ClientConverter();
		c.connectRemote();
	}
	private void connectRemote() throws RemoteException, IOException
	{
		try
		{
			Registry reg=LocateRegistry.getRegistry("localhost", 5678);
			InterfaceConverter h1=(InterfaceConverter)reg.lookup("Wrd");
			BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
			System.out.println("Enter a number: \t");
			String no=br.readLine();
			String ans=h1.convertDigit(no);
			System.out.println("The word representation of the entered digit is:"
+ans);
		}
		catch(NotBoundException|RemoteException e)
		{
			System.out.println("exception" +e);
		}
	}
}
```

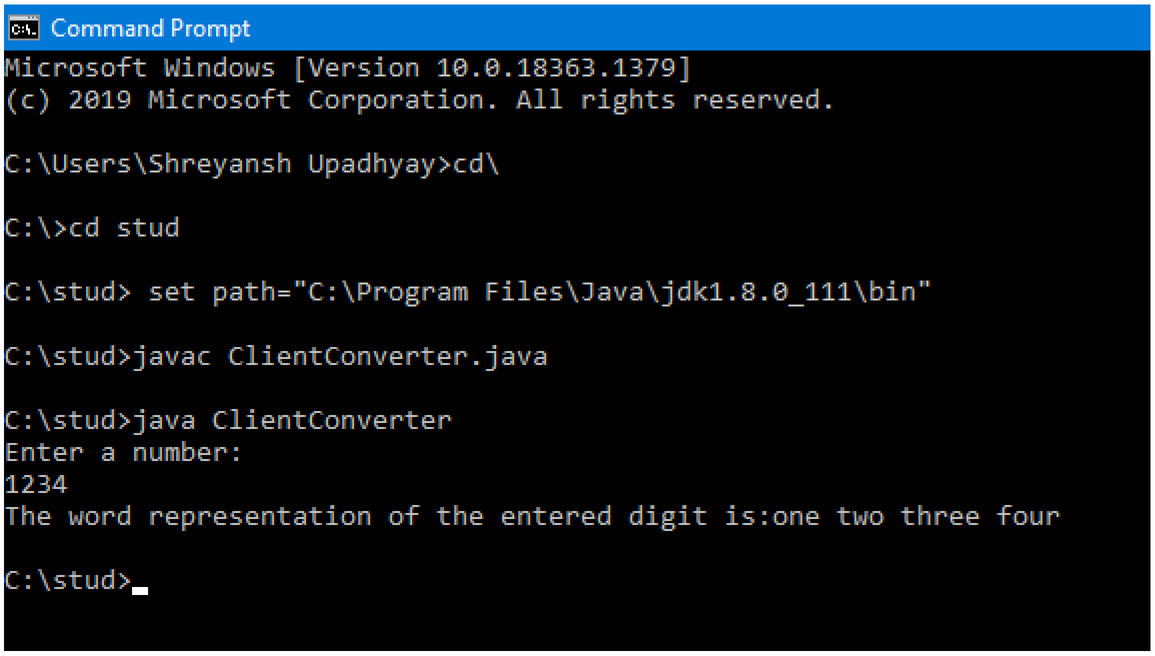
InterfaceConverter.java

```
import java.rmi.*;
public interface InterfaceConverter extends Remote
{
	public String convertDigit(String no) throws RemoteException;
}
```

Output


Command Prompt - java ServerConverter

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud> set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac ServerConverter.java

C:\stud>java ServerConverter
Object Registered...
```


Command Prompt

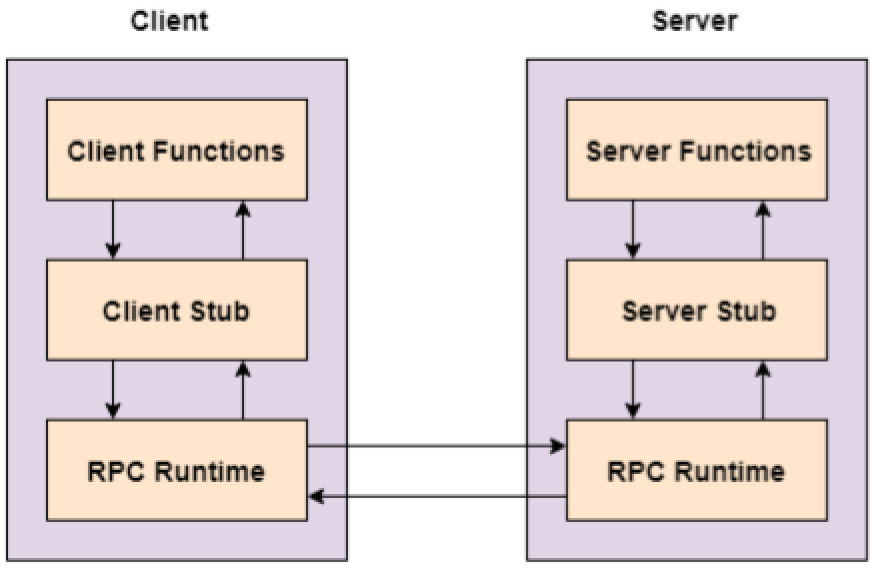
```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud> set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac ClientConverter.java

C:\stud>java ClientConverter
Enter a number:
1234
The word representation of the entered digit is:one two three four

C:\stud>_
```

Practical No.3

Aim: Show the implementation of the Remote Procedure Call

Remote Procedure Call (RPC)

A remote procedure call is an inter process communication technique that is used for client-server-based applications. It is also known as a subroutine call or a function call.
A client has a request message that the RPC translates and sends to the server. This request may be a procedure or a function call to a remote server. When the server receives the request, it sends the required response back to the client. The client is blocked while the server is processing the call and only resumed execution after the server is finished.
The sequence of events in a remote procedure call are given as follows:
• The client stub is called by the client
• The client stub makes a system call to send the message to the server and puts the parameters in the message.
• The message is sent from the client to the server by the client's operating system
• The message is passed to the server stub by the server operating system.
• The parameters are removed from the message by the server stub. • Then, the server procedure is called by the server stub.
A diagram that demonstrates this is as follows:



The following steps take place during a RPC:
1. A client invokes a client stub procedure, passing parameters in the usual way. The client stub resides within the client's own address space.
2. The client stub marshalls(pack) the parameters into a message. Marshalling includes converting the representation of the parameters into a standard format, and copying each parameter into the message.
3. The client stub passes the message to the transport layer, which sends it to the remote server machine.

4. On the server, the transport layer passes the message to a server stub, which demarshalls(unpack) the parameters and calls the desired server routine using the regular procedure call mechanism.

Example 3A: A program to implement simple calculator operations like addition, subtraction, multiplication and division.

RPCServer.java

```
import java.io.*;
import java.net.*;
import java.util.StringTokenizer;
final class RPCServer
{
	DatagramSocket ds;
	DatagramPacket dp;
	String str, methodName, result;
	int val1, val2;
	{
		try
		{
			ds=new DatagramSocket(1200);
			byte b[]=new byte[4096];
			while(true)
			{
				dp=new DatagramPacket(b, b.length);
				ds.receive(dp);
				str=new String(dp.getData(),0,dp.getLength());
				if(str.equalsIgnoreCase("q"))
				{
					System.exit(1);
				}
				else
				{
					StringTokenizer st=new StringTokenizer(str," ");
					int i=0;
					while(st.hasMoreTokens())
					{
						String token=st.nextToken();
						methodName=token;
						val1=Integer.parseInt(st.nextToken());
						val2=Integer.parseInt(st.nextToken());
					}
```

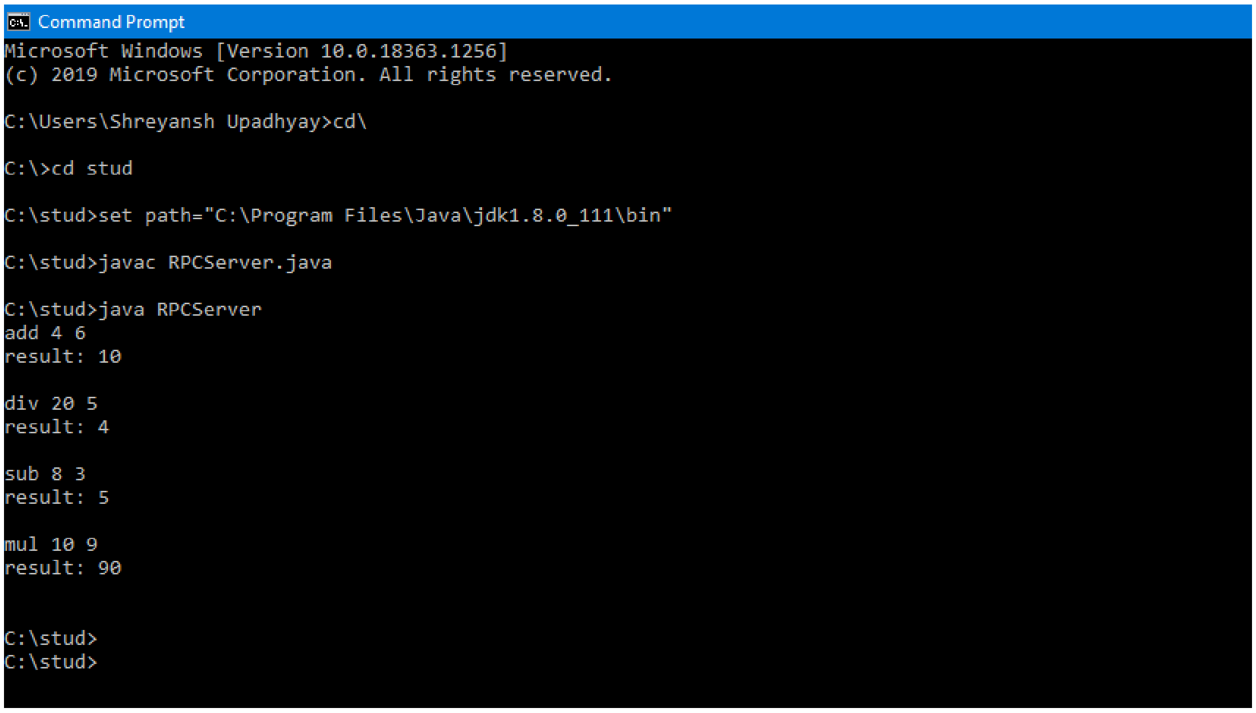
```
                    }
                    System.out.println(str);
                    InetAddress ia=InetAddress.getLocalHost();
                    if(methodName.equalsIgnoreCase("add"))
                    {
                        result=" "+ add(val1, val2);
                    }
                    else if(methodName.equalsIgnoreCase("sub"))
                    {
                        result=" "+ sub(val1, val2);
                    }
                    else if(methodName.equalsIgnoreCase("mul"))
                    {
                        result=" "+ mul(val1, val2);
                    }
                    else if(methodName.equalsIgnoreCase("div"))
                    {
                        result=" "+ div(val1, val2);
                    }
                    byte b1[]=result.getBytes();
                    DatagramSocket ds1=new DatagramSocket();
                    DatagramPacket dp1=new DatagramPacket(b1, b1.length,
InetAddress.getLocalHost(), 1300);
                    System.out.println("result:"+result+"\n");
                    ds1.send(dp1);
                }
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
        }
        public int add(int val1, int val2)
        {
            return val1+val2;
        }
        public int sub(int val3,int val4)
        {
            return val3-val4;
        }
        public int mul(int val3, int val4)
        {
                return val3*val4;
        }
        public int div(int val3, int val4)
        {
                return val3/val4;
        }
        public static void main(String[] args)
        {
                new RPCServer();
        }
}
```

RPCClient.java

```
import java.io.*;
import java.net.*;
class RPCClient
{
        RPCClient()
        {
                try
                {
                        InetAddress ia= InetAddress.getLocalHost();
                        DatagramSocket ds= new DatagramSocket();
                        DatagramSocket ds1= new DatagramSocket(1300);
                        System.out.println("\nRPCClient\n");
                        System.out.println("Enter method name and parameter like add num1
num2\n");
                        while(true)
                        {
                                BufferedReader br= new BufferedReader(new
InputStreamReader(System.in));
                                String str= br.readLine();
                                byte b[]=str.getBytes();
                                DatagramPacket dp=new DatagramPacket(b,b.length,ia,1200);
                                ds.send(dp);
                                dp=new DatagramPacket(b,b.length);
                                ds1.receive(dp);
                                String s=new String(dp.getData(),0,dp.getLength());
```

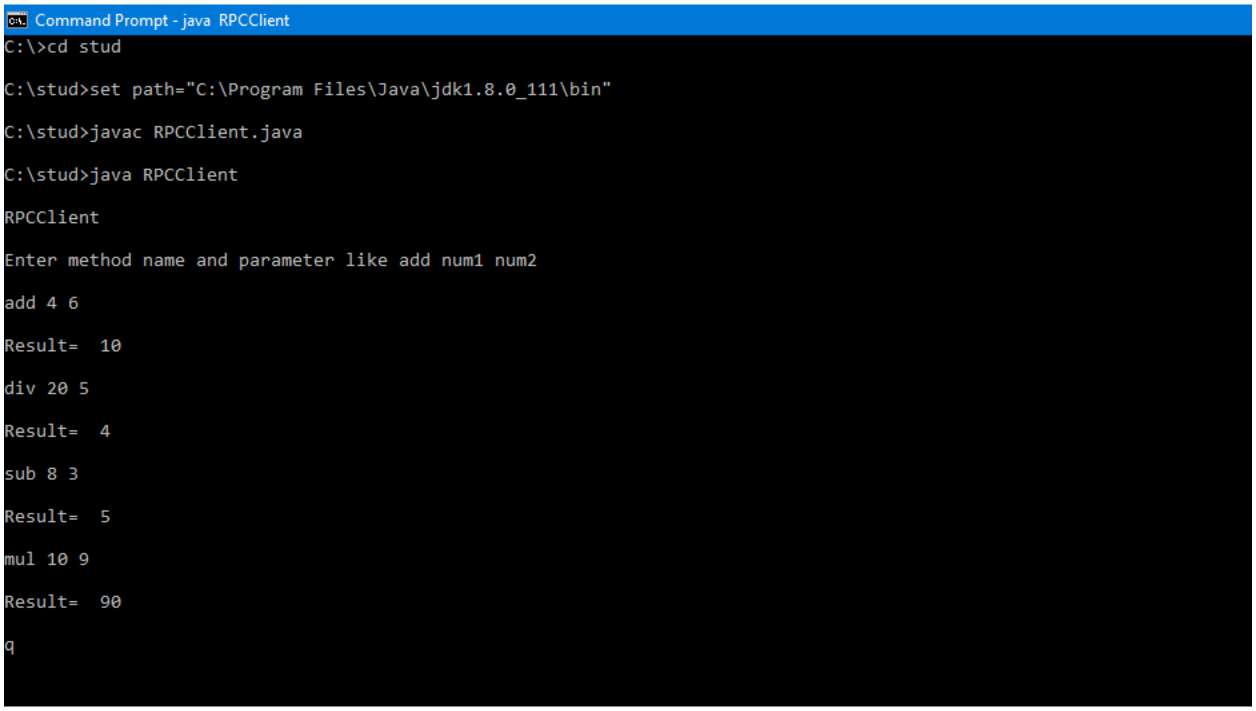
```
                                        System.out.println("\nResult= "+ s +"\n");
                                }
                        }
                        catch(Exception e)
                        {
                                e.printStackTrace();
                        }
                }
                public static void main(String[] agrs)
                {
                        new RPCClient();
                }
}
```

Output



Command Prompt

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac RPCServer.java

C:\stud>java RPCServer
add 4 6
result: 10

div 20 5
result: 4

sub 8 3
result: 5

mul 10 9
result: 90


C:\stud>
C:\stud>
```



Command Prompt - java RPCClient

```
C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac RPCClient.java

C:\stud>java RPCClient

RPCClient

Enter method name and parameter like add num1 num2

add 4 6

Result=  10

div 20 5

Result=  4

sub 8 3

Result=  5

mul 10 9

Result=  90

q
```

Example 3B: A program that finds the square, square root, cube and cube root of the entered number.

RPCServer1.java

```
import java.io.*;
import java.net.*;
import java.util.*;
import java.util.StringTokenizer;
final class RPCServer1
{
        DatagramSocket ds;
        DatagramPacket dp;
        String str, methodName, result;
        int val;
        RPCServer1()
        {
                try
                {
                        ds=new DatagramSocket(1200);
                        byte b[]=new byte[4096];
                        while(true)
                        {
                                dp=new DatagramPacket(b, b.length);
                                ds.receive(dp);
                                str=new String(dp.getData(),0, dp.getLength());
                                if(str.equalsIgnoreCase("q"))
                                {
                                        System.exit(1);
                                }
                                else
                                {
                                        StringTokenizer st=new StringTokenizer(str," ");
                                        int i=0;
                                        while(st.hasMoreTokens())
                                        {
                                                String token=st.nextToken();
                                                methodName=token;
                                                val=Integer.parseInt(st.nextToken());
                                        }
                                }
```

```
                                        System.out.println(str);
                                        InetAddress ia=InetAddress.getLocalHost();
                                        if(methodName.equalsIgnoreCase("square"))
                                        {
                                                result=" "+ square(val);
                                        }
                                        else if(methodName.equalsIgnoreCase("sqroot"))
                                        {
                                                result=" "+ sqroot(val);
                                        }
                                        else if(methodName.equalsIgnoreCase("cube"))
                                        {
                                                result=" "+ cube(val);
                                        }
                                        else if(methodName.equalsIgnoreCase("cbroot"))
                                        {
                                                result=" "+ cbroot(val);
                                        }
                                        byte b1[]=result.getBytes();
                                        DatagramSocket ds1=new DatagramSocket();
                                        DatagramPacket dp1=new DatagramPacket(b1, b1.length,
InetAddress.getLocalHost(), 1300);
                                        System.out.println("result:"+result+"\n");
                                        ds1.send(dp1);
                                }
                        }
                        catch(Exception e)
                        {
                                e.printStackTrace();
                        }
                }
                public double square(int val)
                {
                        return Math.pow(val,2);
                }
                public double sqroot(int val)
                {
                        return Math.sqrt(val);
                }
                public double cube(int val)
```

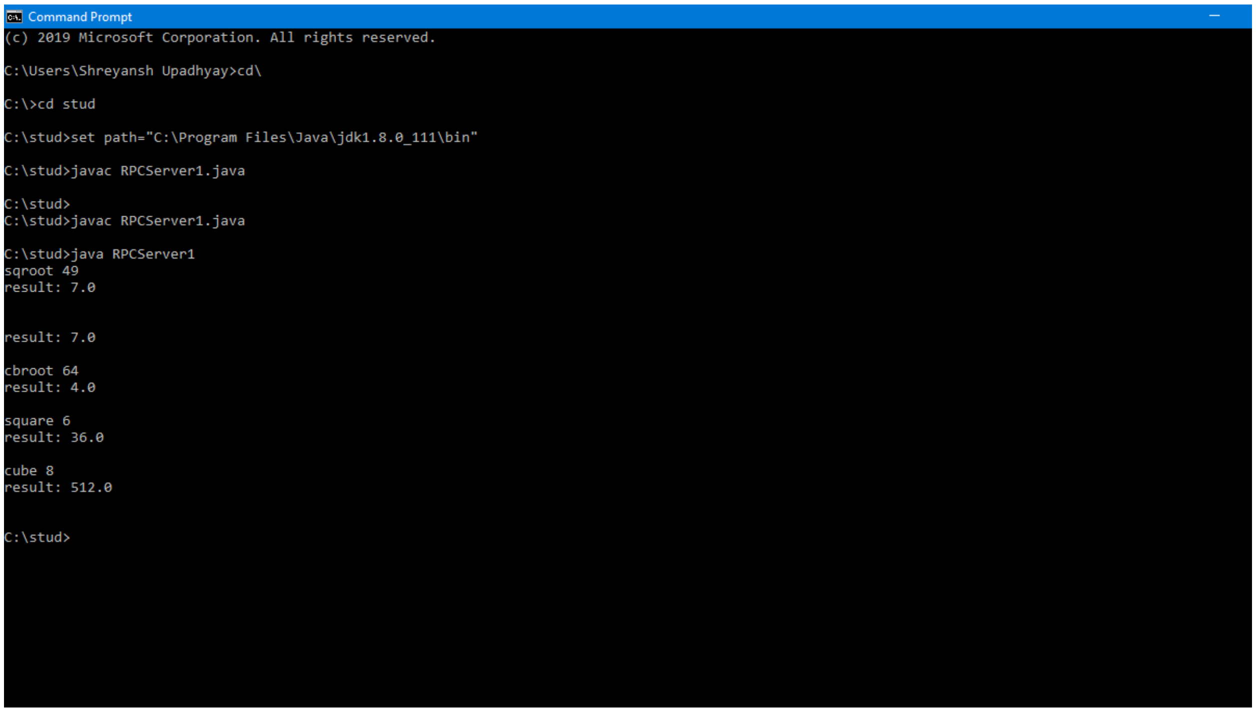
```
	{
		return Math.pow(val,3);
	}
	public double cbroot(int val)
	{
		return Math.cbrt(val);
	}
	public static void main(String[] args)
	{
		new RPCServer1();
	}
}
```

RCPClient1.java

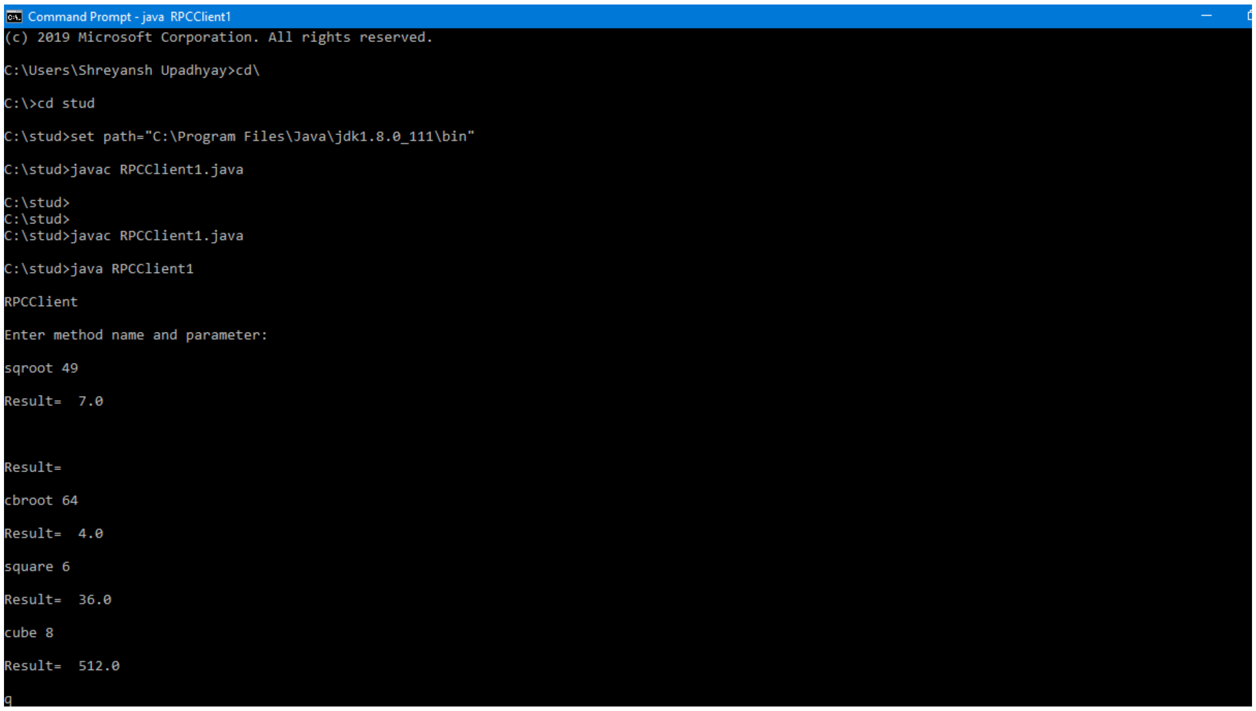
```
import java.io.*;
import java.net.*;
import java.util.*;
class RPCClient1
{
	RPCClient1()
	{
		try
		{
			InetAddress ia= InetAddress.getLocalHost();
			DatagramSocket ds= new DatagramSocket();
			DatagramSocket ds1= new DatagramSocket(1300);
			System.out.println("\nRPCClient\n");
			System.out.println("Enter method name and parameter:\n");
			while(true)
			{
				BufferedReader br= new BufferedReader(new
InputStreamReader(System.in));
				String str= br.readLine();
				byte b[]=str.getBytes();
				DatagramPacket dp=new DatagramPacket(b,b.length,ia,1200);
				ds.send(dp);
				dp=new DatagramPacket(b,b.length);
				ds1.receive(dp);
				String s=new String(dp.getData(),0,dp.getLength());
```

```
                        System.out.println("\nResult= "+ s +"\n");
                    }
                }
                catch(Exception e)
                {
                    e.printStackTrace();
                }
            }
            public static void main(String[] agrs)
            {
                new RPCClient1();
            }
        }
```

Output



```
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac RPCServer1.java

C:\stud>
C:\stud>javac RPCServer1.java

C:\stud>java RPCServer1
sqroot 49
result: 7.0


result: 7.0

cbroot 64
result: 4.0

square 6
result: 36.0

cube 8
result: 512.0


C:\stud>
```



```
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac RPCClient1.java

C:\stud>
C:\stud>
C:\stud>javac RPCClient1.java

C:\stud>java RPCClient1

RPCClient

Enter method name and parameter:

sqroot 49

Result=  7.0


Result=

cbroot 64

Result=  4.0

square 6

Result=  36.0

cube 8

Result=  512.0

q
```

Practical No. 4

Aim: Show the implementation of Web Services

Web Services

A Web Service is a software program that uses XML to exchange information with other software via common internet protocols. In a simple sense, Web Services are a way of interacting with objects over the Internet.

A web service is
• Language Independent.
• Protocol Independent.
• Platform Independent.
• It assumes a stateless service architecture.
• Scalable (e.g. multiplying two numbers together to an entire customer-relationship management system).
• Programmable (encapsulates a task).
• Based on XML (open, text-based standard).
• Self-describing (metadata for access and use).
• Discoverable (search and locate in registries)- ability of applications and developers to search for and locate desired Web services through registries. This is based on UDDI.

Key Web Service Technologies
1. XML- Describes only data. So, any application that understands XML-regardless of the application's programming language or platform has the ability to format XML in a variety of ways (well-formed or valid). 2. SOAP- Provides a communication mechanism between services and applications. 3. WSDL- Offers a uniform method of describing web services to other programs. 4. UDDI- Enables the creation of searchable Web services registries.

Web Services Limitations
1. SOAP, WSDL, UDDI- require further development. 2. Interoperability. 3. Royalty fees. 4. Too slow for use in high-performance situations. 5. Increase traffic on networks. 6. The lack of security standards for Web services. 7. The standard procedure for describing the quality (i.e. levels of performance, reliability, security etc.) of particular Web services – management of Web services. 8. The standards that drive Web services are still in draft form (always will be in refinement).
23

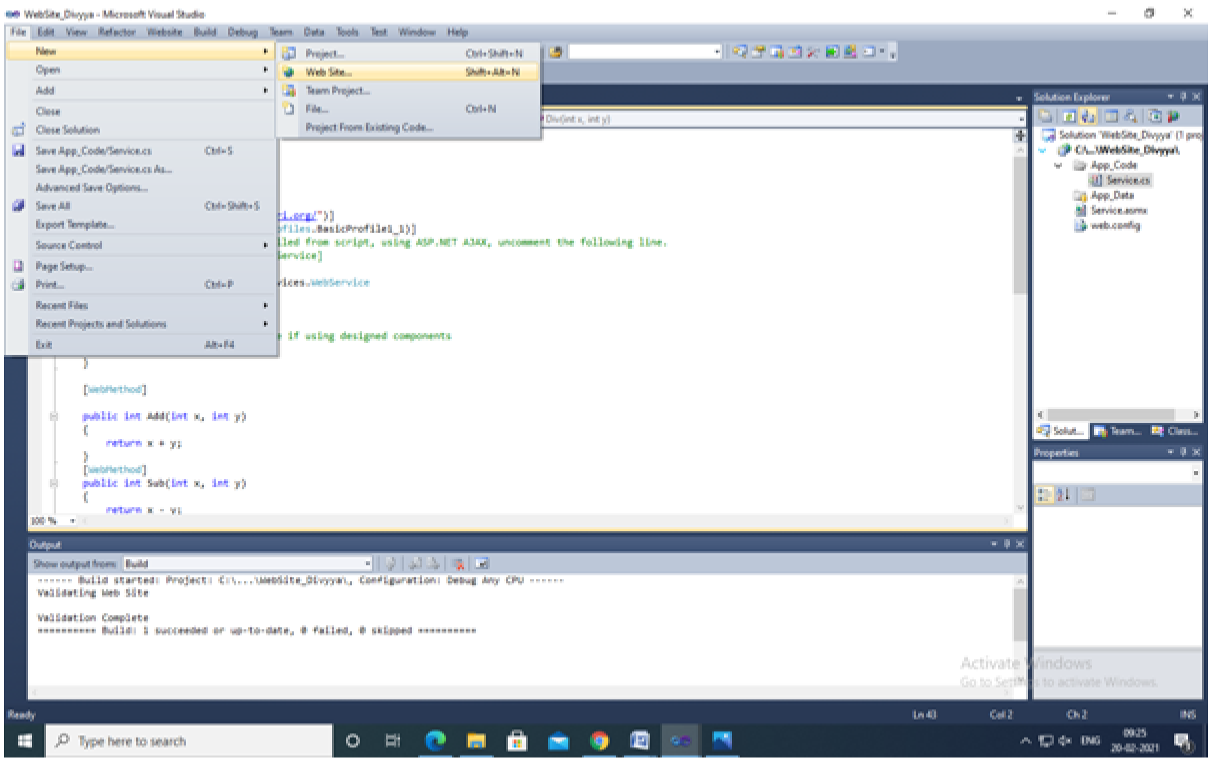
9. Some vendors want to retain their intellectual property rights to certain Web services standards.

A web service can perform almost any kind of task
1. Web Portal- A web portal might obtain top news headlines from an associated press web service. 2. Weather Reporting- It can use as Weather reporting web service to display weather information in your personal website. 3. Stock Quote- It can display latest update of Share market with Stock Quote on your web site. 4. News Headline- You can display latest news update by using News Headline Web Service in your website. 5. You can make your own web service and let others use it. For example, you can make Free SMS Sending Service with footer with your companies' advertisement, so whosoever uses this service indirectly advertises your company. You can apply your ideas in N no. of ways to take advantage of it.

Step 1

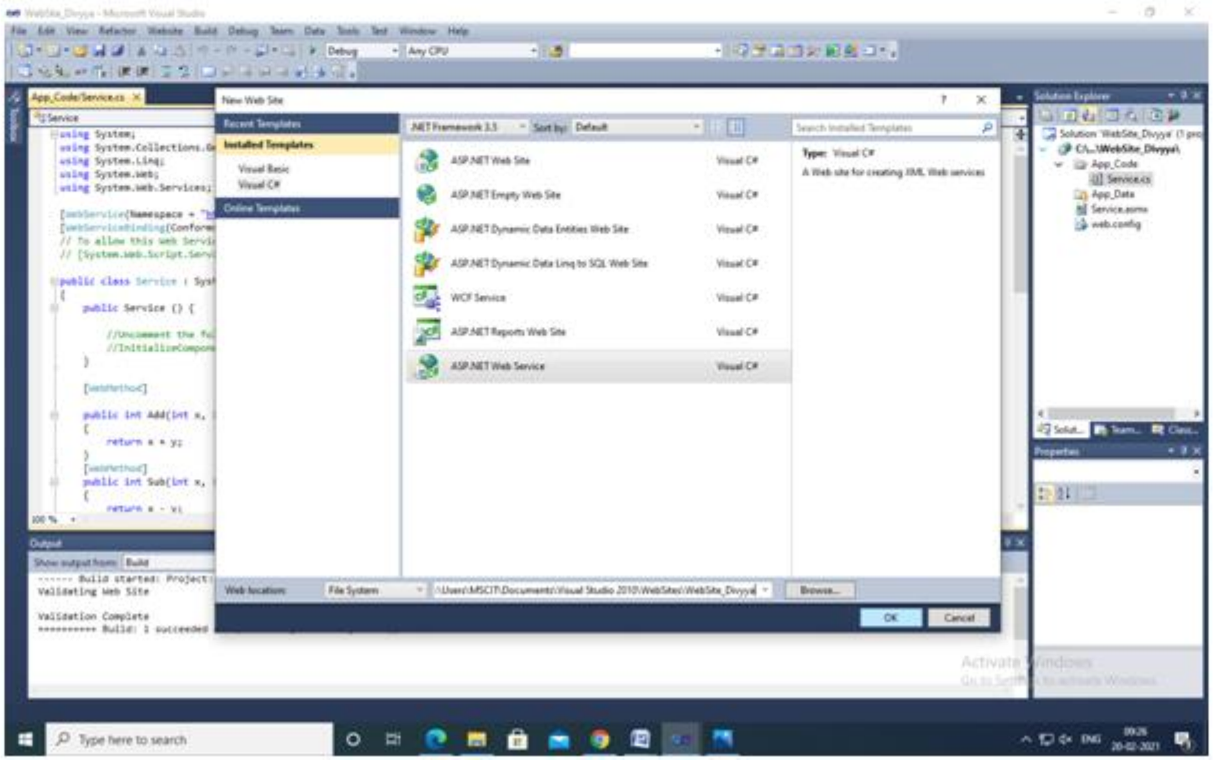
Create a ASP.NET Web service

Click on File->New->Website->ASP.NET Web Service and name the Web Service.



```
        [WebMethod]
        public int Add(int x, int y)
        {
            return x + y;
        }
        [WebMethod]
        public int Sub(int x, int y)
        {
            return x - y;
```

```
------ Build started: Project: C:\...\WebSite_Divyya\, Configuration: Debug Any CPU ------
Validating Web Site

Validation Complete
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```



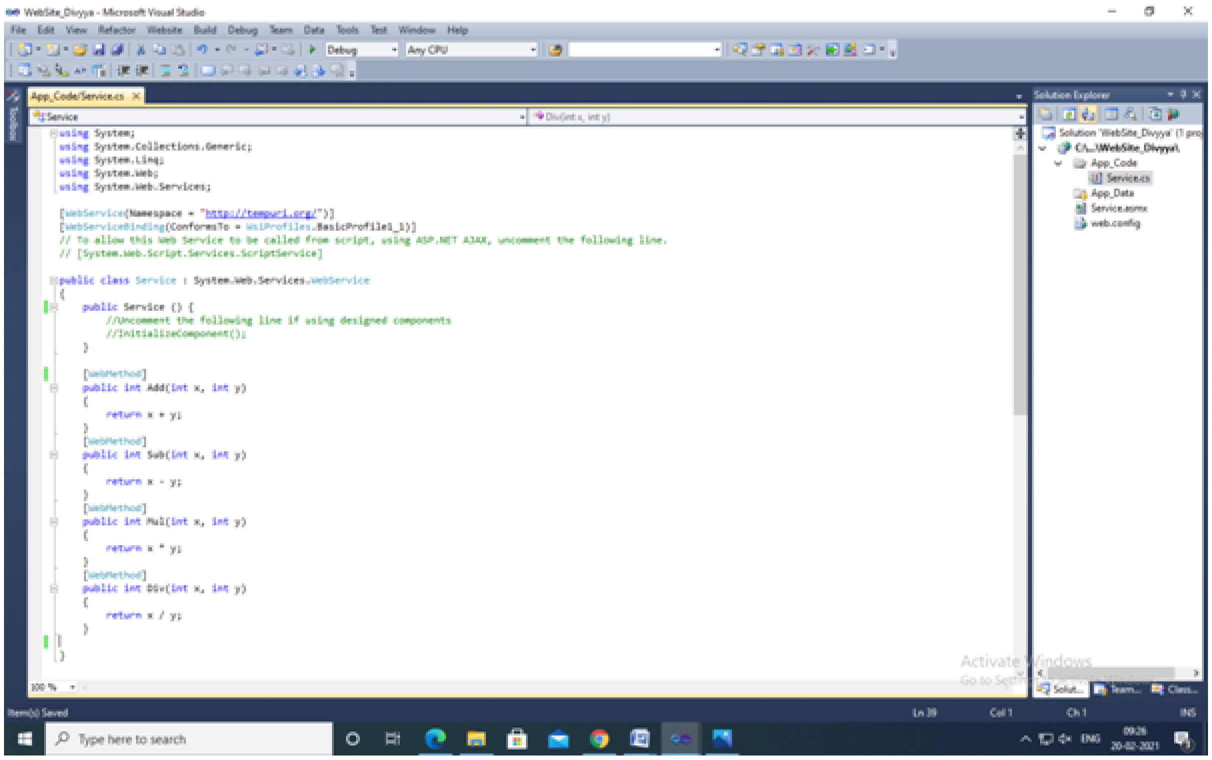
.NET Framework 3.5 — Sort by: Default

Type: Visual C#

A Web site for creating XML Web services

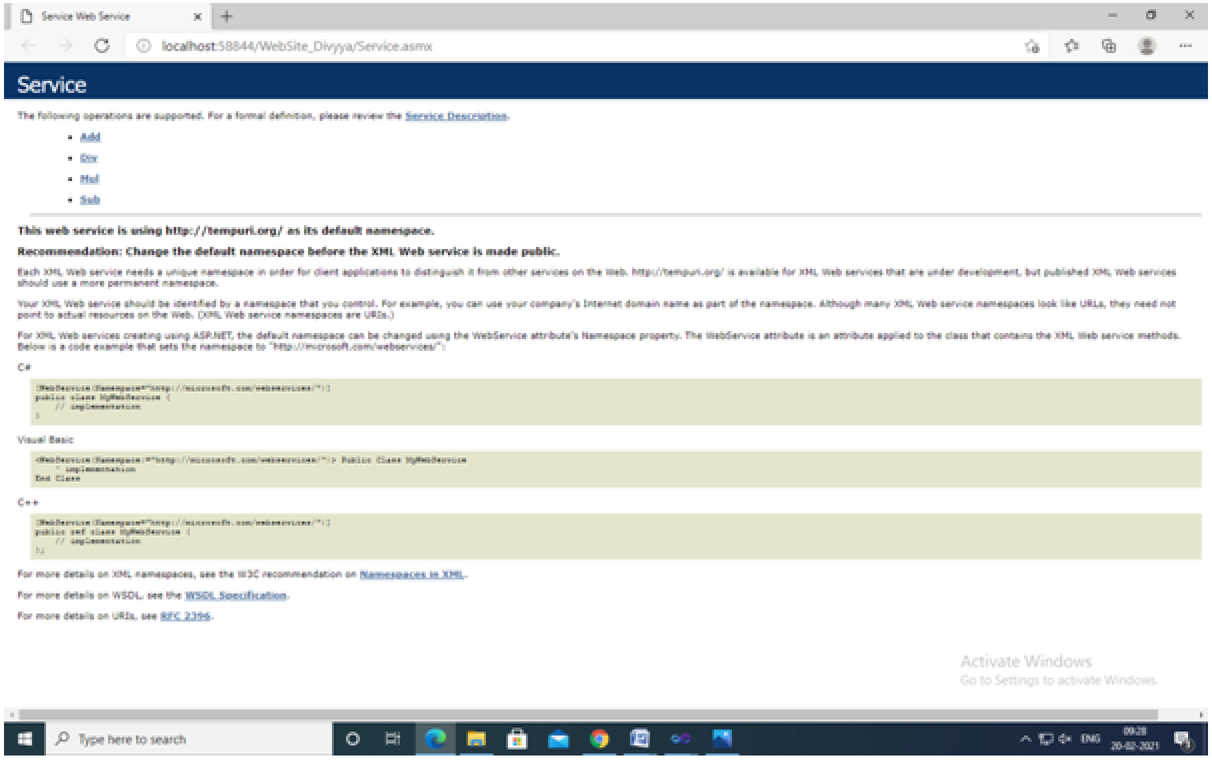
Web location: File System — \Users\MSCIT\Documents\Visual Studio 2010\WebSites\WebSite_Divyya

Step2: Type the following code



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
// [System.Web.Script.Services.ScriptService]

public class Service : System.Web.Services.WebService
{
    public Service () {
        //Uncomment the following line if using designed components
        //InitializeComponent();
    }

    [WebMethod]
    public int Add(int x, int y)
    {
        return x + y;
    }
    [WebMethod]
    public int Sub(int x, int y)
    {
        return x - y;
    }
    [WebMethod]
    public int Mul(int x, int y)
    {
        return x * y;
    }
    [WebMethod]
    public int Div(int x, int y)
    {
        return x / y;
    }
}
}
```

Step 3: Debug the Web Service and Copy the path



localhost:58844/WebSite_Divyya/Service.asmx

Service

The following operations are supported. For a formal definition, please review the Service Description.

- Add
- Div
- Mul
- Sub

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

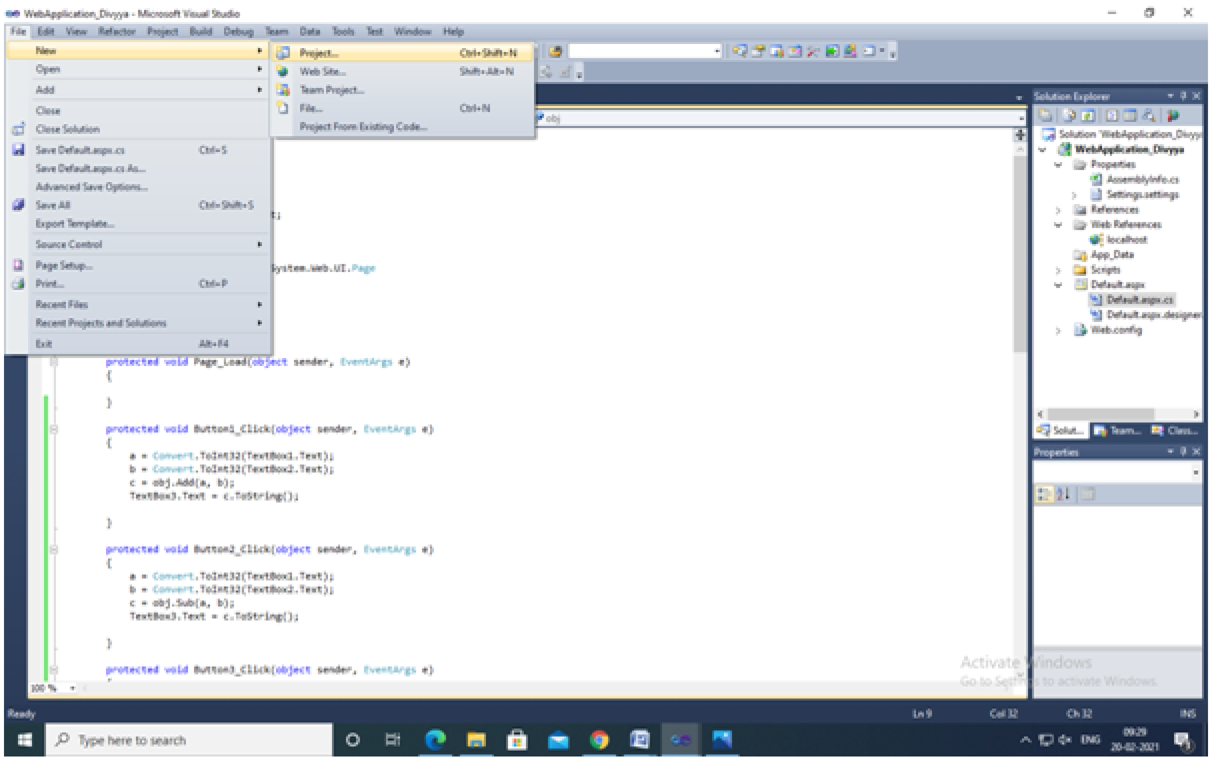
```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

For more details on XML namespaces, see the W3C recommendation on Namespaces in XML.

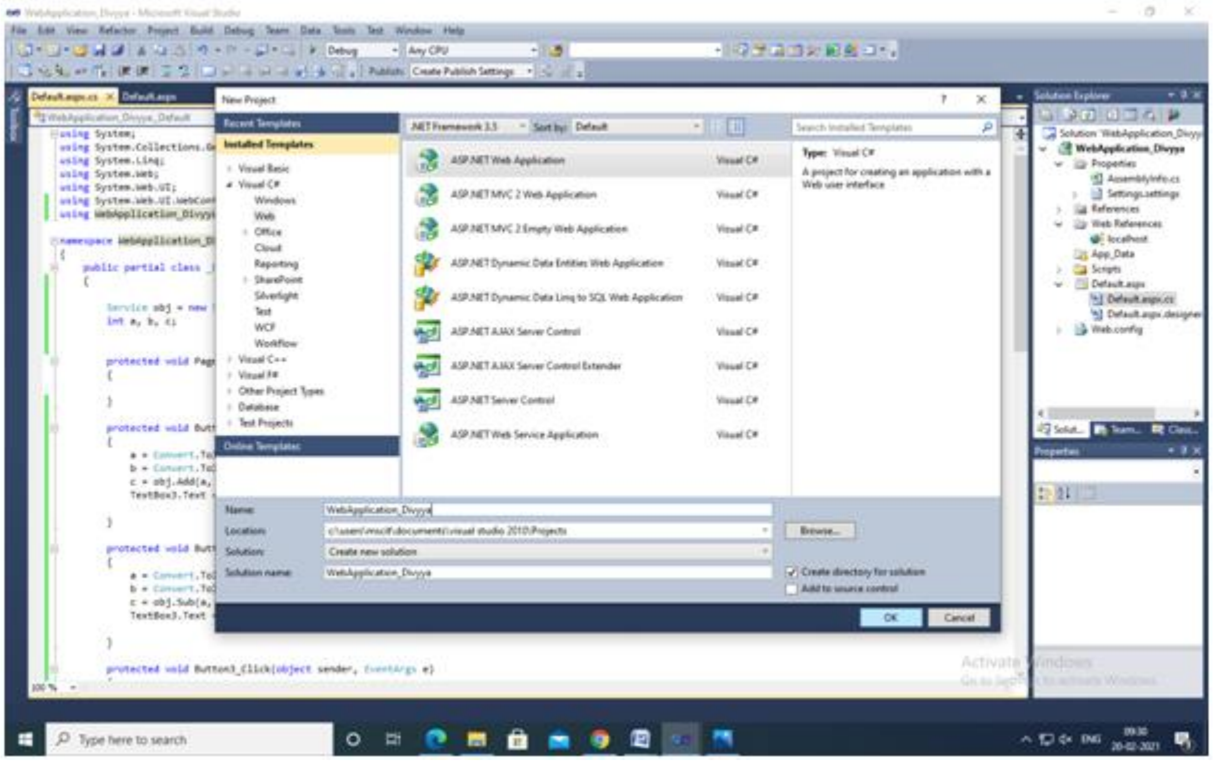
For more details on WSDL, see the WSDL Specification.

For more details on URIs, see RFC 2396.

Step 4: Create a ASP.NET We Application. Click on File->New->Project->Web Application and name the Web Application.



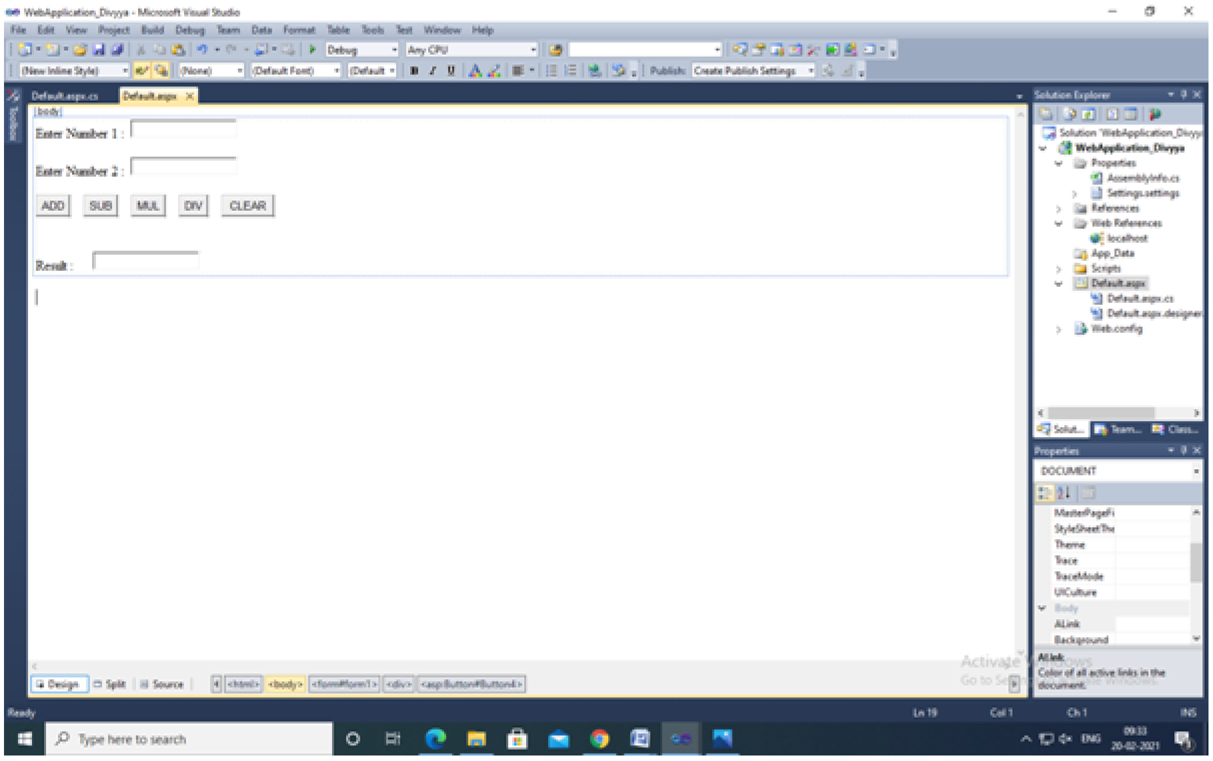
```
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Add(a, b);
            TextBox3.Text = c.ToString();

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Sub(a, b);
            TextBox3.Text = c.ToString();

        }

        protected void Button3_Click(object sender, EventArgs e)
```


New Project
Recent Templates
Installed Templates
Visual Basic
Visual C#
Windows
Web
Office
Cloud
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
Visual C++
Visual F#
Other Project Types
Database
Test Projects
Online Templates
.NET Framework 3.5
Sort by: Default
ASP.NET Web Application Visual C#
ASP.NET MVC 2 Web Application Visual C#
ASP.NET MVC 2 Empty Web Application Visual C#
ASP.NET Dynamic Data Entities Web Application Visual C#
ASP.NET Dynamic Data Linq to SQL Web Application Visual C#
ASP.NET AJAX Server Control Visual C#
ASP.NET AJAX Server Control Extender Visual C#
ASP.NET Server Control Visual C#
ASP.NET Web Service Application Visual C#
Browse...
OK
Cancel

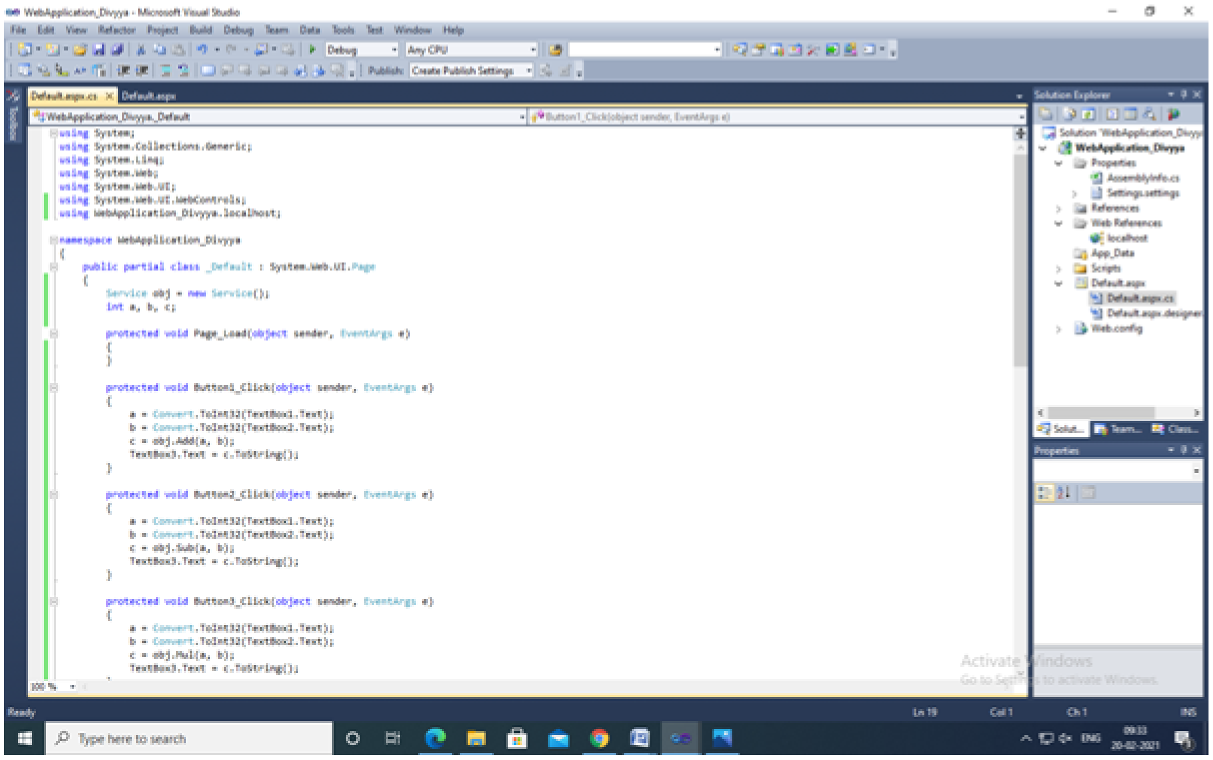
Type: Visual C#
A project for creating an application with a Web user interface

Name: WebApplication_Divyya
Location: c:\users\mscit\documents\visual studio 2010\Projects
Solution: Create new solution
Solution name: WebApplication_Divyya
[✓] Create directory for solution
[ ] Add to source control

Step 5: Design a simple Calculator

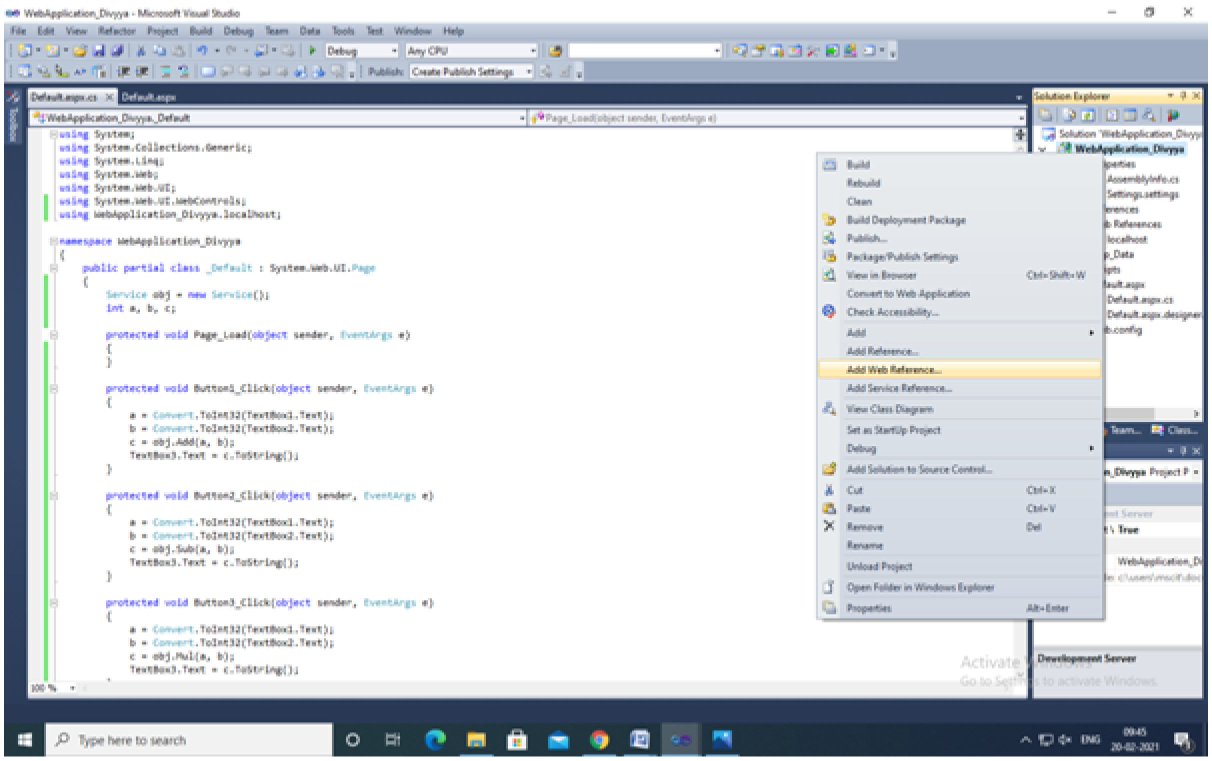


Step 6: Type the following code

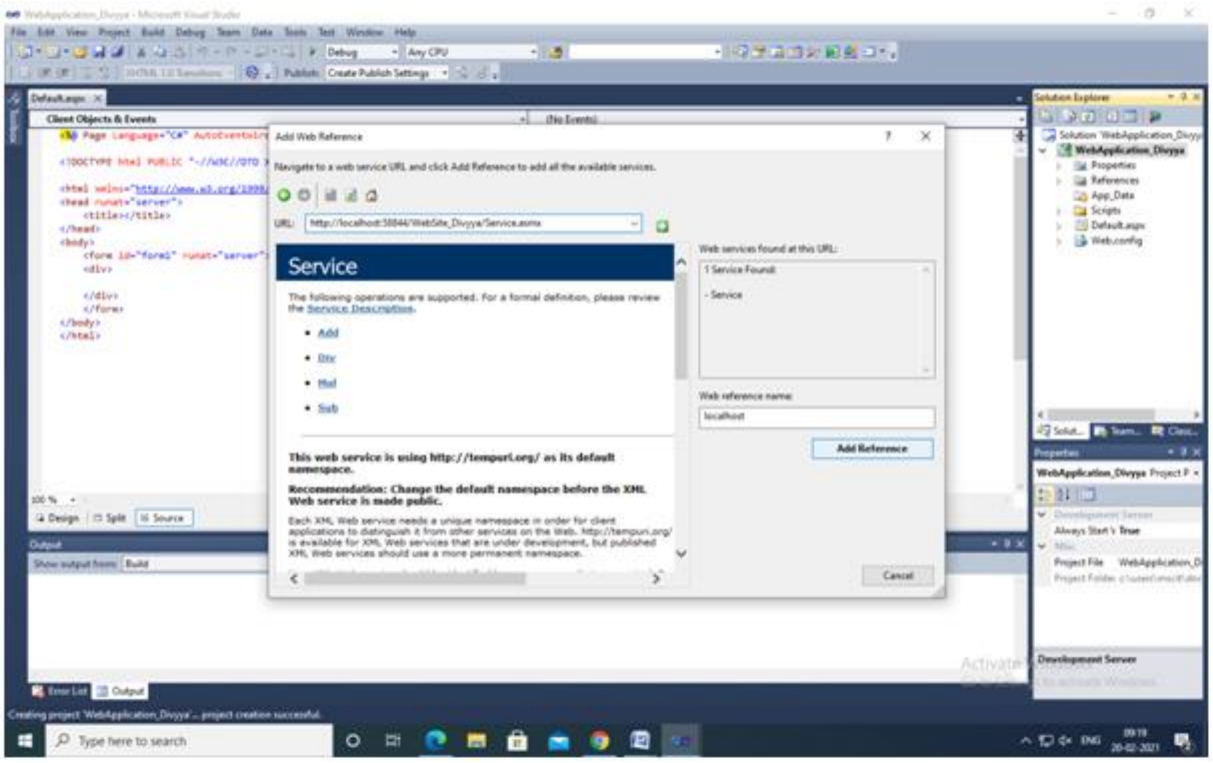


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WebApplication_Divyya.localhost;

namespace WebApplication_Divyya
{
    public partial class _Default : System.Web.UI.Page
    {
        Service obj = new Service();
        int a, b, c;

        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Add(a, b);
            TextBox3.Text = c.ToString();
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Sub(a, b);
            TextBox3.Text = c.ToString();
        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Mul(a, b);
            TextBox3.Text = c.ToString();
```

Step 7: Right click on the web application and click ->Add Web reference and copy the path and click on add Reference.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using WebApplication_Divyya.localhost;

namespace WebApplication_Divyya
{
    public partial class _Default : System.Web.UI.Page
    {
        Service obj = new Service();
        int a, b, c;

        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Add(a, b);
            TextBox3.Text = c.ToString();
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Sub(a, b);
            TextBox3.Text = c.ToString();
        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            a = Convert.ToInt32(TextBox1.Text);
            b = Convert.ToInt32(TextBox2.Text);
            c = obj.Mul(a, b);
            TextBox3.Text = c.ToString();
```



Navigate to a web service URL and click Add Reference to add all the available services.

URL: http://localhost:38844/WebSite_Divyya/Service.asmx

The following operations are supported. For a formal definition, please review the Service Description.

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Web services found at this URL:

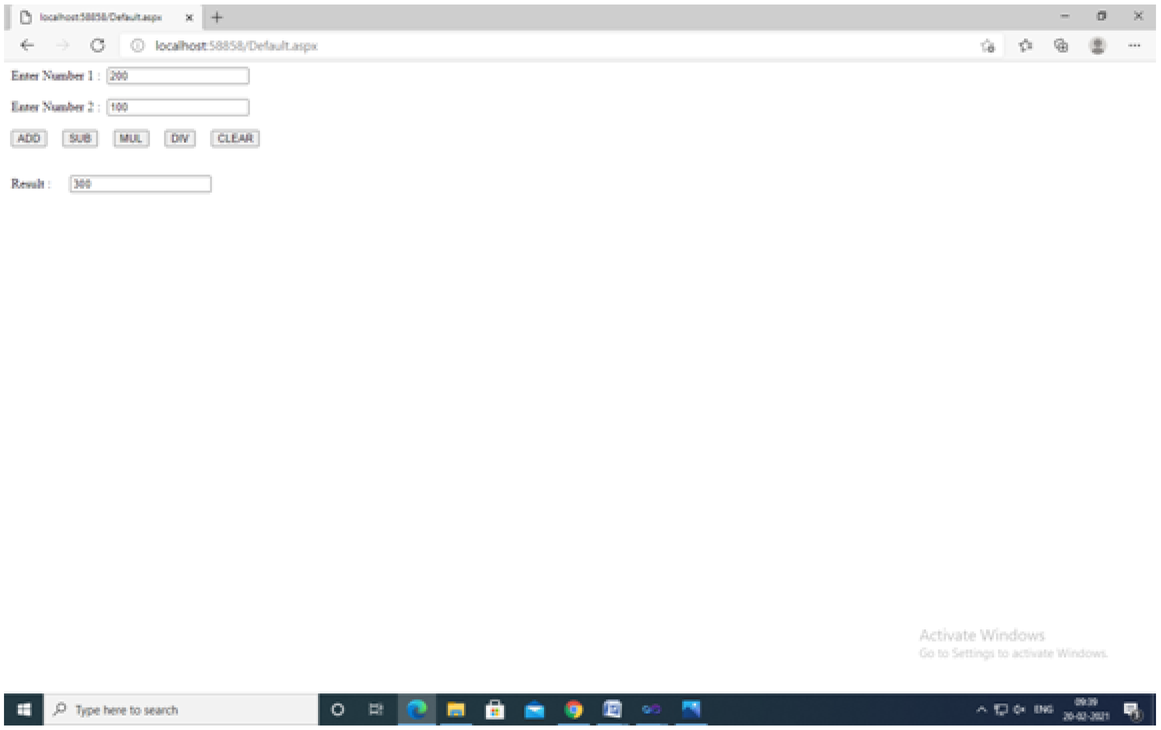
1 Service Found:

- Service

Web reference name: localhost

Creating project 'WebApplication_Divyya'... project creation successful.

Step 8: Debug the web application


localhost:58858/Default.aspx
localhost:58858/Default.aspx

Enter Number 1 : 200

Enter Number 2 : 100

ADD SUB MUL DIV CLEAR

Result : 300

Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
09:39
20-02-2021

Practical No. 5

Aim: Write a program to execute any one Mutual Exclusion Algorithm

Mutual exclusion is a concurrency control property which is introduced to prevent race conditions. It is the requirement that a process cannot enter its critical section while another concurrent process is currently present or executing in its critical section i.e. only one process is allowed to execute the critical section at any given instance of time.

Distributed mutual exclusion algorithms must deal with unpredictable message delays and incomplete knowledge of the system state.
Three basic approaches for distributed mutual exclusion:
1. Token based approach 2. Non-token-based approach 3. Quorum based approach

Token-based approach: • A unique token is shared among the sites. • A site is allowed to enter its CS if it possesses the token. • Mutual exclusion is ensured because the token is unique. • Example: Suzuki-Kasami's Broadcast Algorithm

Non-token-based approach: • Two or more successive rounds of messages are exchanged among the sites to determine which site will enter the CS next. • Example: Lamport's algorithm, Ricart–Agrawala algorithm

Quorum based approach: • Each site requests permission to execute the CS from a subset of sites (called a quorum). • Any two quorums contain a common site. • This common site is responsible to make sure that only one request executes the CS at any time. • Example: Maekawa's Algorithm

Requirements of Mutual Exclusion Algorithms:
1. No Deadlock: Two or more site should not endlessly wait for any message that will never arrive.
32

2. No Starvation: Every site who wants to execute critical section should get an opportunity to execute it in finite time. Any site should not wait indefinitely to execute critical section while other site is repeatedly executing critical section
3. Fairness: Each site should get a fair chance to execute critical section. Any request to execute critical section must be executed in the order they are made i.e Critical section execution requests should be executed in the order of their arrival in the system.
4. Fault Tolerance: In case of failure, it should be able to recognize it by itself in order to continue functioning without any disruption.

TokenServer.java

```
import java.net.*;
public class TokenServer
{
        public static void main(String args[]) throws Exception
        {
                while(true)
                {
                        Server sr=new Server();
                        sr.recPort(8000);
                        sr.recData();
                }
        }
}
class Server
{
        boolean hasToken=false;
        boolean sendData=false;
        int recport;
        void recPort(int recport)
        {
                this.recport=recport;
        }
        void recData() throws Exception
        {
                byte b[]=new byte[256];
                DatagramSocket ds;
                DatagramPacket dp;
                String str;
                ds=new DatagramSocket(recport);
                dp=new DatagramPacket(b,b.length);
                ds.receive(dp);
                ds.close();
                str=new String(dp.getData(),0,dp.getLength());
                System.out.println("The message is:" +str);
        }
}
```

TokenClient1.java

```
import java.io.*;
import java.net.*;
public class TokenClient1
{
        public static void main(String args[]) throws Exception
        {
                InetAddress lclhost;
                BufferedReader br;
                String str="";
                TokenClient12 tkcl, tkser;
                boolean hasToken;
                boolean setSendData;
                while(true)
                {
                        lclhost=InetAddress.getLocalHost();
                        tkcl=new TokenClient12(lclhost);
                        tkser=new TokenClient12(lclhost);
                        tkcl.setSendPort(9004);
                        tkcl.setRecPort(8002);
                        lclhost=InetAddress.getLocalHost();
                        tkser.setSendPort(9000);
                        if(tkcl.hasToken==true)
                        {
                                System.out.println("Do you want to enter Data?-->YES/NO");
                                br=new BufferedReader(new InputStreamReader(System.in));
                                str=br.readLine();
                                if(str.equalsIgnoreCase("yes"))
                                {
                                        System.out.println("Reay to send");
                                        tkser.setSendData=true;
                                        tkser.sendData();
                                        tkser.setSendData=false;
                                }
                                else if(str.equalsIgnoreCase("no"))
                                {
                                        tkcl.sendData();
                                        tkcl.recData();
                                }
                        }
                        else
                        {
                                System.out.println("ENTERING RECEIVING MODE...");
                                tkcl.recData();
                        }
                }
        }
}
class TokenClient12
{
```

```
InetAddress lclhost;
int sendport, recport;
boolean hasToken=true;
boolean setSendData=false;
TokenClient12 tkcl, tkser;
TokenClient12(InetAddress lclhost)
{
        this.lclhost=lclhost;
}
void setSendPort(int sendport)
{
        this.sendport=sendport;
}
void setRecPort(int recport)
{
        this.recport=recport;
}
void sendData() throws Exception
{
        BufferedReader br;
        String str="Token";
        DatagramSocket ds;
        DatagramPacket dp;
        if(setSendData=true)
        {
                System.out.println("Enter the Dataata");
                br=new BufferedReader(new InputStreamReader(System.in));
                str="Client One..."+br.readLine();
                System.out.println("Now Sending...");
                System.out.println("Data Sent");
        }
        ds=new DatagramSocket(sendport);
        dp=new DatagramPacket(str.getBytes(), str.length(),lclhost,sendport-1000);
        ds.send(dp);
        ds.close();D
        setSendData=false;
        hasToken=false;
}
void recData() throws Exception
{
```

```
                String msgstr;
                byte b[]=new byte[256];
                DatagramSocket ds;
                DatagramPacket dp;
                ds=new DatagramSocket(recport);
                dp=new DatagramPacket(b,b.length);
                ds.receive(dp);
                ds.close();
                msgstr=new String(dp.getData(),0,dp.getLength());
                System.out.println("The Data is:" +msgstr);
                if(msgstr.equals("Token"))
                {
                        hasToken=true;
                }
        }
}
```

TokenClient2.java

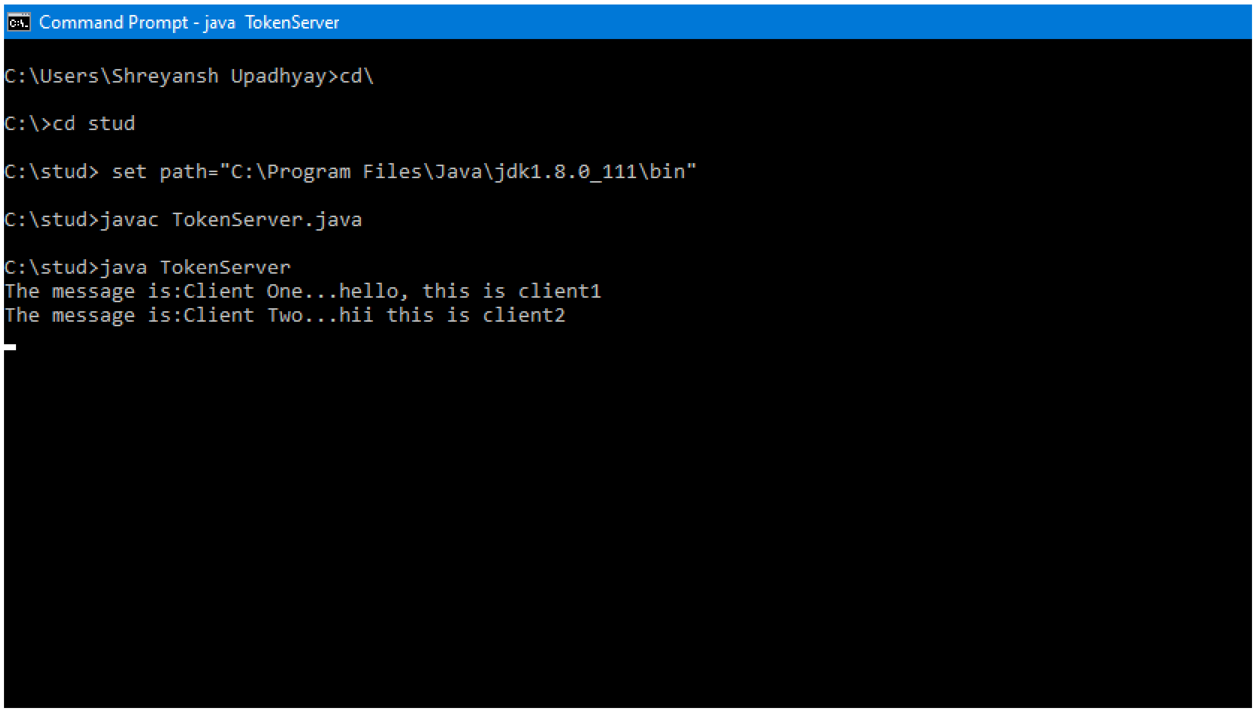
```
import java.io.*;
import java.net.*;
public class TokenClient2
{
        static boolean setSendData;
        static boolean hasToken;
        public static void main(String args[]) throws Exception
        {
                InetAddress lclhost;
                BufferedReader br;
                String str1;
                TokenClient21 tkcl;
                TokenClient21 ser;
                while(true)
                {
                        lclhost=InetAddress.getLocalHost();
                        tkcl=new TokenClient21(lclhost);
                        tkcl.setRecPort(8004);
                        tkcl.setSendPort(9002);
                        lclhost=InetAddress.getLocalHost();
                        ser=new TokenClient21(lclhost);
```

```
					ser.setSendPort(9000);
					if(hasToken==true)
					{
						System.out.println("Do you want to enter the data?-->YES/NO");
						br=new BufferedReader(new InputStreamReader(System.in));
						str1=br.readLine();
						if(str1.equalsIgnoreCase("yes"))
						{
							System.out.println("Ready to Send...");
							ser.setSendData=true;
							ser.sendData();
							ser.setSendData=false;
						}
						else if(str1.equalsIgnoreCase("no"))
						{
							tkcl.sendData();
							tkcl.recData();
						}
					}
					else
					{
						System.out.println("Entering Receiving Mode...");
						tkcl.recData();
						hasToken=true;
					}
				}
		}
}
class TokenClient21
{
		InetAddress lclhost;
		int sendport, recport;
		boolean setSendData=false;
		boolean hasToken=false;
		TokenClient21 tkcl;
		TokenClient21 ser;
		TokenClient21(InetAddress lclhost)
		{
				this.lclhost=lclhost;
		}
```

```
void setSendPort(int sendport)
{
	this.sendport=sendport;
}
void setRecPort(int recport)
{
	this.recport=recport;
}
void sendData() throws Exception
{
	BufferedReader br;
	String str="Token";
	DatagramSocket ds;
	DatagramPacket dp;
	if(setSendData=true)
	{
		System.out.println("Enter the Dataata");
		br=new BufferedReader(new InputStreamReader(System.in));
		str="Client Two..."+br.readLine();
		System.out.println("Now Sending...");
		System.out.println("Data Sent!!!");
	}
	ds=new DatagramSocket(sendport);
	dp=new DatagramPacket(str.getBytes(), str.length(),lclhost,sendport-1000);
	ds.send(dp);
	ds.close();
	setSendData=false;
	hasToken=false;
}
void recData() throws Exception
{
	String msgstr;
	byte b[]=new byte[256];
	DatagramSocket ds;
	DatagramPacket dp;
	ds=new DatagramSocket(recport);
	dp=new DatagramPacket(b,b.length);
	ds.receive(dp);
	ds.close();
	msgstr=new String(dp.getData(),0,dp.getLength());
```

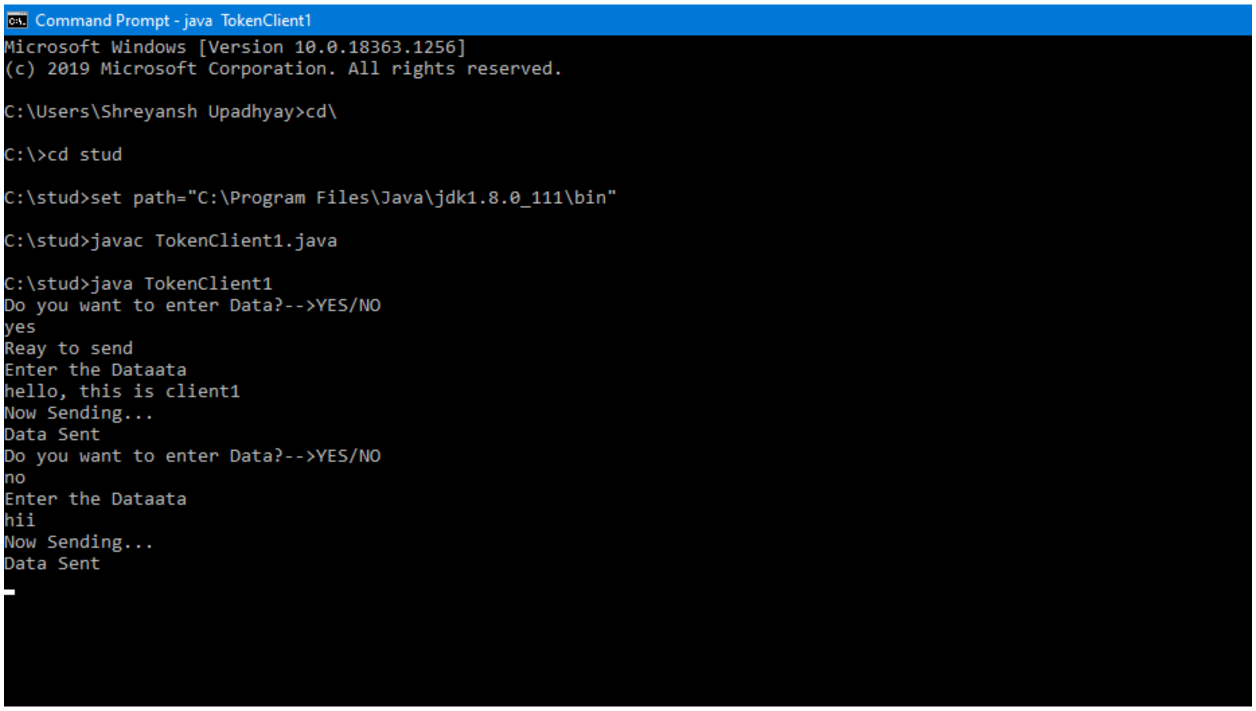
```
                System.out.println("The Data is:" +msgstr);
                if(msgstr.equals("Token"))
                {
                        hasToken=true;
                }
        }
}
```

Output

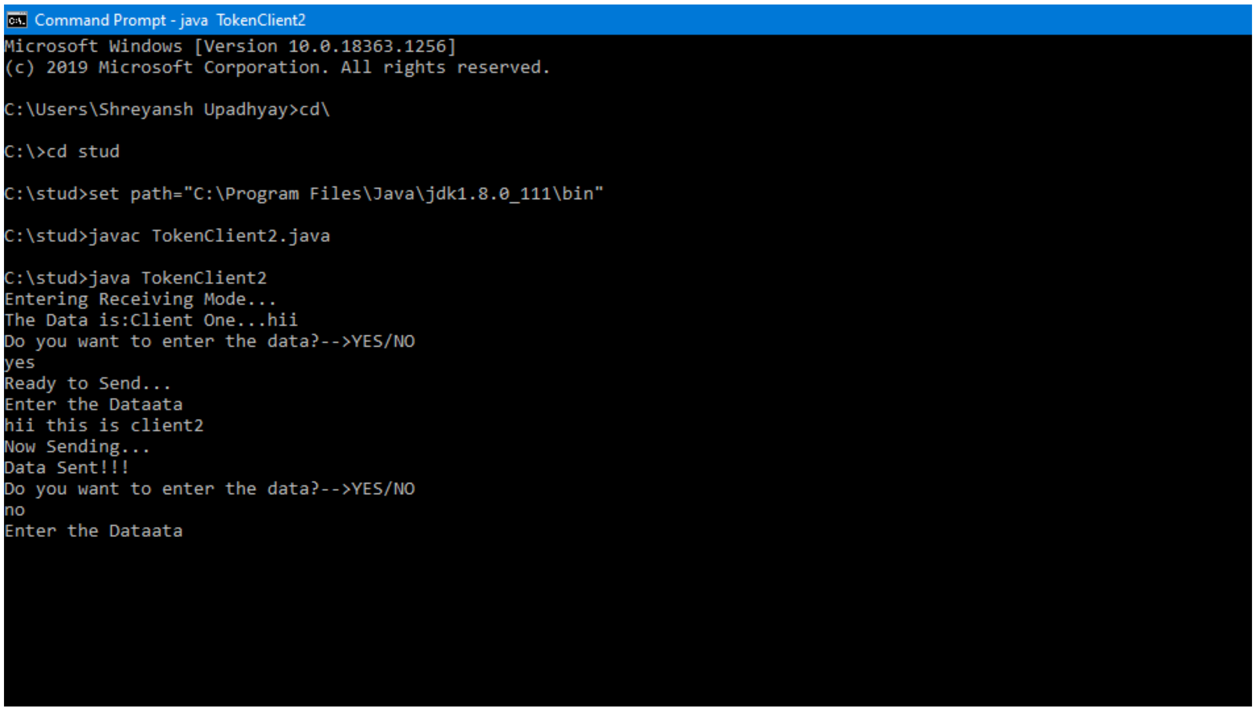


Command Prompt - java TokenServer

```
C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud> set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TokenServer.java

C:\stud>java TokenServer
The message is:Client One...hello, this is client1
The message is:Client Two...hii this is client2
```


Command Prompt - java TokenClient1

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TokenClient1.java

C:\stud>java TokenClient1
Do you want to enter Data?-->YES/NO
yes
Reay to send
Enter the Dataata
hello, this is client1
Now Sending...
Data Sent
Do you want to enter Data?-->YES/NO
no
Enter the Dataata
hii
Now Sending...
Data Sent
```


Command Prompt - java TokenClient2

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TokenClient2.java

C:\stud>java TokenClient2
Entering Receiving Mode...
The Data is:Client One...hii
Do you want to enter the data?-->YES/NO
yes
Ready to Send...
Enter the Dataata
hii this is client2
Now Sending...
Data Sent!!!
Do you want to enter the data?-->YES/NO
no
Enter the Dataata
```

Practical No. 6

Aim: Write a program to implement any one Election Algorithm

Election Algorithm

Many distributed algorithms require the election of a special coordinator process; one that has a special role, or initiates something, or monitors something. Often it doesn't matter who the special process is, but one and only one must be elected, and it can't be known in advance who it will be. On a low-level you can think about the monitor in a token ring as an example.
The assumptions of these algorithms are that every process can be uniquely identified (by IP address, for example), and that each process can find out the id of the other processes. What the processes don't know is which processes are up and which are down at any given point in time.

Bully algorithm

Garcia-Molina, 1982. The process with the highest identity always becomes the coordinator. When a process P sees that the coordinator is no longer responding to requests it initiates an election by sending ELECTION messages to all processes whose id is higher than its own. If no one responds to the messages then P is the new coordinator. If one of the higher-ups responds, it takes over and P doesn't have to worry anymore.
When a process receives an ELECTION message it sends a response back saying OK. It then holds its own election (unless it is already holding one). Eventually there is only one process that has not given up and that is the new coordinator. It is also the one with the highest number currently running. When the election is done the new coordinator sends a COORDINATOR message to everyone informing them of the change.
If a process which was down comes back up, it immediately holds an election. If this process had previously been the coordinator it will take this role back from whoever is doing it currently (hence the name of the algorithm).

Ring algorithm

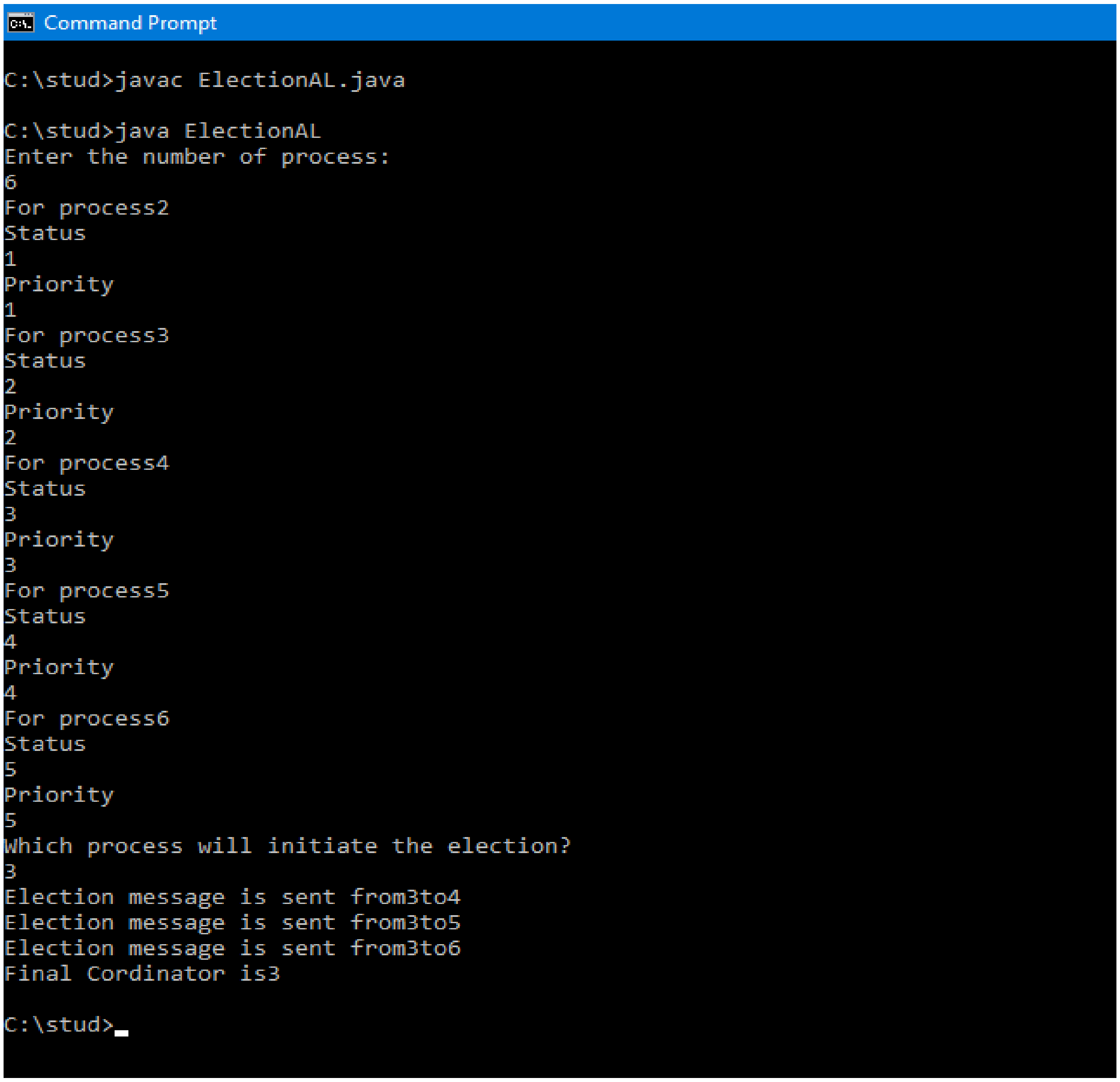
Assumes that processes are logically ordered in some fashion, and that each process knows the order and who is coordinator. No token is involved. When a process notices that the coordinator is not responding it sends an ELECTION message with its own id to its downstream neighbor. If that neighbor doesn't respond it sends it to its neighbor's neighbor, etc. Each station that receives the ELECTION message adds its own id to the list. When the message circulates back to the originator it selects the highest id in the list and sends a COORDINATOR message announcing the new coordinator. This message circulates once and is removed by the originator.
If two elections are held simultaneously (say because two different processes notice simultaneously that the coordinator is dead) then each comes up with the same list and elects the same coordinator. Some time is wasted, but nothing is really hurt by this.

ElectionAL.java

```
import java.io.*;
import java.util.Scanner;
class ElectionAL
{
        static int n;
        static int pro[]=new int[100];
```

```
	static int sta[]=new int[100];
	static int co;
	public static void main(String args[])
	{
		System.out.println("Enter the number of process:");
		Scanner in=new Scanner(System.in);
		n=in.nextInt();
		int i;
		for(i=1; i<n; i++)
		{
			System.out.println("For process" +(i+1)+"");
			System.out.println("Status");
			sta[i]=in.nextInt();
			System.out.println("Priority");
			pro[i]=in.nextInt();
		}
		System.out.println("Which process will initiate the election?");
		int ele=in.nextInt();
		elect(ele);
		System.out.println("Final Cordinator is" +co);
	}
	static void elect(int ele)
	{
		ele=ele-1;
		co=ele+1;
		for(int i=0; i<n; i++)
		{
			if(pro[ele]<pro[i])
			{
				System.out.println("Election message is sent
from"+(ele+1)+"to"+(i+1));
				if(sta[i]==1)
					elect(i+1);
			}
		}
	}
}
```

Output



Command Prompt

```
C:\stud>javac ElectionAL.java

C:\stud>java ElectionAL
Enter the number of process:
6
For process2
Status
1
Priority
1
For process3
Status
2
Priority
2
For process4
Status
3
Priority
3
For process5
Status
4
Priority
4
For process6
Status
5
Priority
5
Which process will initiate the election?
3
Election message is sent from3to4
Election message is sent from3to5
Election message is sent from3to6
Final Cordinator is3

C:\stud>_
```

Practical No. 7

Aim: Show the implementation of any one clock synchronization algorithm

Clock Synchronization Algorithm

Clock synchronization is a method of synchronizing clock values of any two nodes in a distributed system with the use of external reference clock or internal clock value of the node. During the synchronization, many factors effect on a network. As discussed above, these factors need to be considered before correcting actual clock value. Based on the approach, clock synchronization algorithms are divided as Centralized Clock Synchronization and Distributed Clock Synchronization Algorithm.

Distributed algorithm:

• There is particular time server.

• The processor periodically reach an agreement on the clock value by averaging the time of neighbor clock and its local clock.

• This can be used if no UTC receiver exist (no external synchronization is needed). Only internal synchronization is performed.

• Processes can run different machine and no global clock to judge which event happened first.

SCServer.java

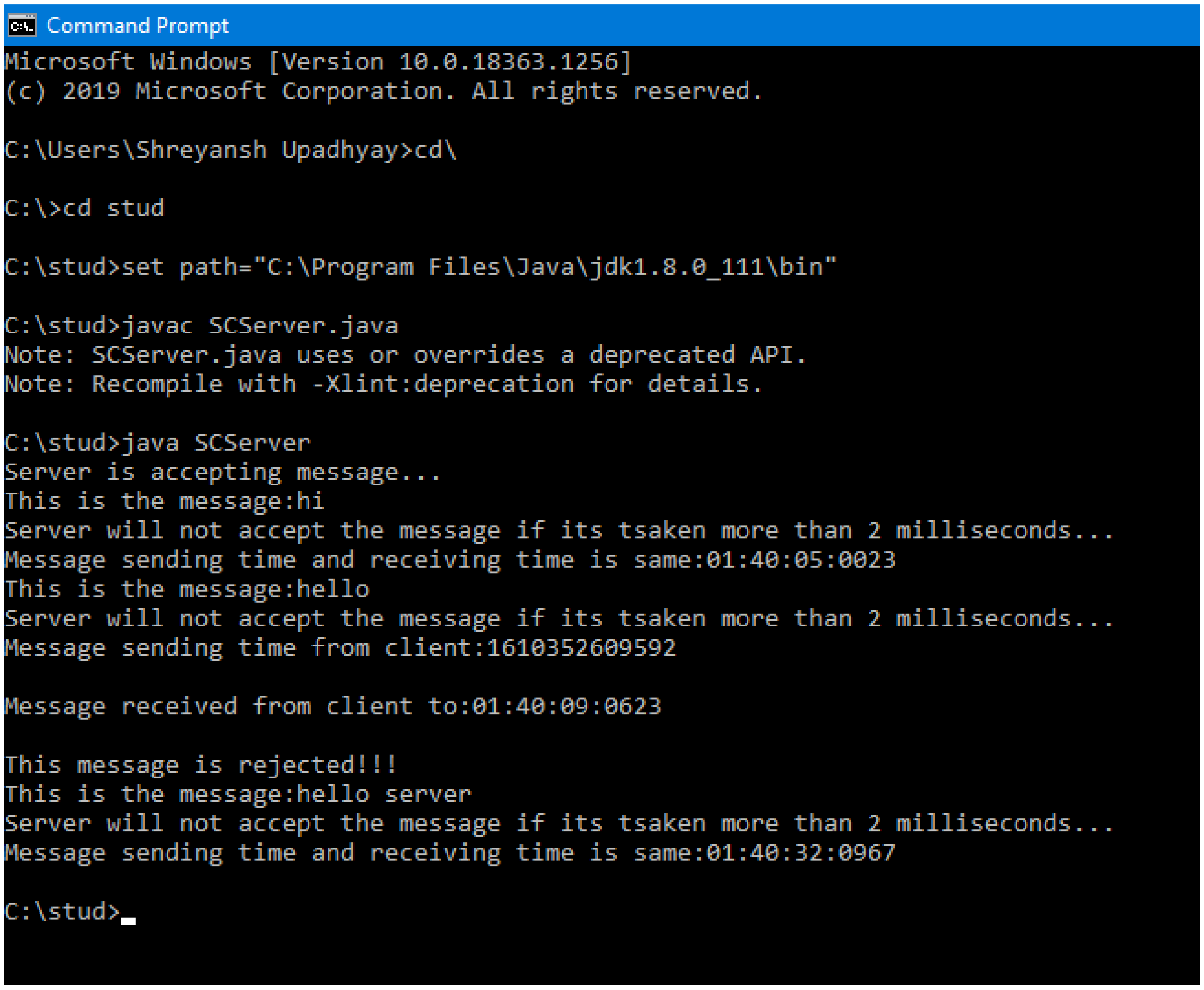
```
import java.net.*;
import java.io.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
public class SCServer
{
	public static void main(String args[])
	{
		try
		{
			ServerSocket ss=new ServerSocket(8001);
			System.out.println("Server is accepting message...");
			Socket s=ss.accept();
			DataInputStream in=new DataInputStream(s.getInputStream());
			String receive;
			DateFormat dateFormat=new SimpleDateFormat("hh:mm:ss:SSSS");
			while((receive=in.readLine())!=null)
			{
				String[] message=receive.split(",");
				if(message[0].equals("stop") || message[0].equals("Stop") || 
message[0].equals("STOP"))
				break;
```

```
                                        Date date=new Date();
                                        Long clienttime=Long.parseLong(message[1]);
                                        Long timeMilli=date.getTime();
                                        Long requiretime=timeMilli-clienttime;
                                        System.out.println("This is the message:" +message[0]);
                                        System.out.println("Server will not accept the message if its tsaken
more than 2 milliseconds...");
                                        if(clienttime.equals(timeMilli))
                                        {
                                                String strDate=dateFormat.format(timeMilli);
                                                System.out.println("Message sending time and receiving
time is same:" +strDate);
                                        }
                                        else if(requiretime>2)
                                        {
                                                String clienttim=dateFormat.format(clienttime);
                                                System.out.println("Message sending time from client:"
+clienttime+"\n");
                                                String strDate=dateFormat.format(timeMilli);
                                                System.out.println("Message received from client to:"
+strDate+"\n");
                                                System.out.println("This message is rejected!!!");
                                        }
                                        else
                                        {
                                                String clinttim = dateFormat.format(clienttime);
                                                System.out.println("Messange sending time from client:"+
clinttim+"\n");
                                                String strDate = dateFormat.format(timeMilli);
                                                System.out.println("Messange received from client
to:"+strDate+"\n");
                                                System.out.println("This message is accepted.");
                                        }
                                }
                                in.close();
                        }
                        catch(Exception e)
                        {
                                System.out.println(e.toString());
                        }
                }
}
```

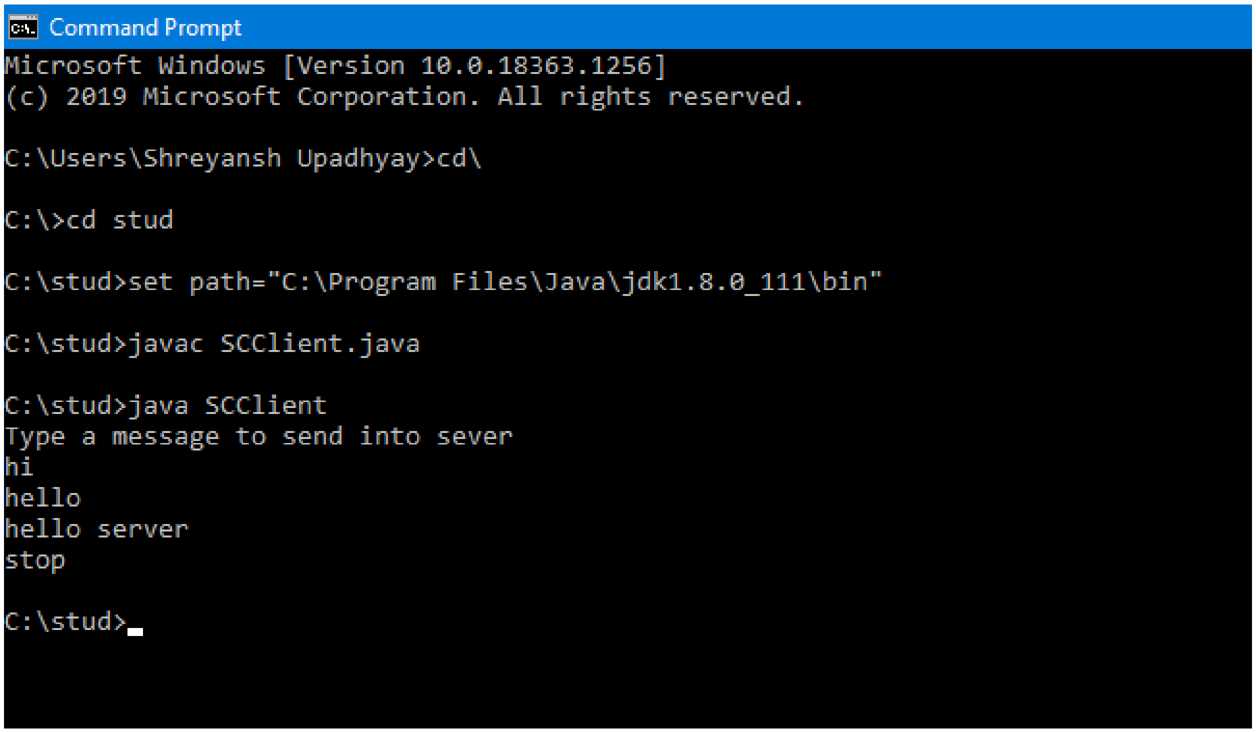
SCClient.java

```
import java.net.*;
import java.io.*;
import java.util.Date;
public class SCClient
{
	public static void main(String args[])
	{
		try
		{
			Socket cs = new Socket("LocalHost",8001);
			BufferedReader infu = new BufferedReader(new
InputStreamReader(System.in));
			DataOutputStream ot = new DataOutputStream(cs.getOutputStream());
			String send;
			System.out.println("Type a message to send into sever");
			while((send = infu.readLine()) != null )
			{
				Date date = new Date();
				long timeMilli = date.getTime();
				String t=String.valueOf(timeMilli);
				ot.writeBytes(send+","+t+"\n");
				if(send.equals("stop") || send.equals("STOP") ||
send.equals("Stop"))
				break;
			}
			infu.close();
			ot.close();
			cs.close();
		}
		catch(Exception e)
		{
			System.out.println(e.toString());
		}
	}
}
```

Output


Command Prompt

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac SCServer.java
Note: SCServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\stud>java SCServer
Server is accepting message...
This is the message:hi
Server will not accept the message if its tsaken more than 2 milliseconds...
Message sending time and receiving time is same:01:40:05:0023
This is the message:hello
Server will not accept the message if its tsaken more than 2 milliseconds...
Message sending time from client:1610352609592

Message received from client to:01:40:09:0623

This message is rejected!!!
This is the message:hello server
Server will not accept the message if its tsaken more than 2 milliseconds...
Message sending time and receiving time is same:01:40:32:0967

C:\stud>_
```



Command Prompt

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac SCClient.java

C:\stud>java SCClient
Type a message to send into sever
hi
hello
hello server
stop

C:\stud>
```

Practical No. 8

Aim: Write a program to implement two phase commit protocol

Two Phase Commit protocol follows following steps:

Step 1: The coordinator sends a VOTE_REQUEST message to all the participants.

Step 2: When a participant receives a VOTE_REQUEST message , it returns either a VOTE_COMMIT message to the coordinator telling the coordinator that it is prepared to cmooit or a VOTE_ABORT message.

Step 3: The coordinator collects all votes from the participants. If all the participants voted to commit the transaction then it sends GLOBAL_COMMIT message to all the participants. Even if one participant have voted to abort the transaction then it sends GLOBAL_ABORT message.

Step 4: Each participant that voted for a commit waits for the final reaction by the coordinator. If the participant receives a GLOBAL_COMMIT message it locally commits transcation else it abort the local transaction.

TPCServer.java

```
import java.io.*;
import java.net.*;
public class TPCServer
{
	public static void main(String args[]) throws Exception
	{
		BufferedReader br;
		InetAddress lclhost;
		lclhost=InetAddress.getLocalHost();
		Server ser=new Server(lclhost);
		System.out.println("Server in sending mode...");
		ser.setSendPort(9000);
		ser.setRecPort(8001);
		System.out.println("Send request data to client1...");
		br=new BufferedReader(new InputStreamReader(System.in));
		String s=br.readLine();
		System.out.println("Data is:" +s);
		ser.sendData();
		System.out.println("Waiting for response from client1...");
		ser.recData();
		ser.setSendPort(9002);
		ser.setRecPort(8003);
		System.out.println("Sending request data to client2...");
		br=new BufferedReader(new InputStreamReader(System.in));
```

```
                String s1=br.readLine();
                System.out.println("Data is" +s1);
                ser.sendData();
                System.out.println("waiting for response from client2...");
                ser.recData();
                ser.setSendPort(9000);
                ser.sendData();
        }
}
class Server
{
        InetAddress lclhost;
        int sendPort, recPort;
        int ssend=0;
        int scounter=0;
        Server(InetAddress lclhost)
        {
                this.lclhost=lclhost;
        }
        public void setSendPort(int sendPort)
        {
                this.sendPort=sendPort;
        }
        public void setRecPort(int recPort)
        {
                this.recPort=recPort;
        }
        public void sendData() throws Exception
        {
                DatagramSocket ds;
                DatagramPacket dp;
                String data="";
                if(scounter<2 && ssend<2)
                {
                        data="VOTE_REQUEST";
                }
                if(scounter<2 && ssend>1)
                {
                        data="GLOBAL_ABORT";
                        data=data+"TRANSACTION_ABORTED";
```

```
		}
		if(scounter==2 && ssend>1)
		{
			data="GLOBAL_COMMIT";
			data=data+"TRANSACTION_COMMITED";
		}
		ds=new DatagramSocket(sendPort);
		dp=new DatagramPacket(data.getBytes(),data.length(),lclhost,sendPort-1000);
		ds.send(dp);
		ds.close();
		ssend++;
	}
	public void recData() throws Exception
	{
		byte buf[]=new byte[256];
		DatagramPacket dp=null;
		DatagramSocket ds=null;
		String msgStr="";
		try
		{
			ds=new DatagramSocket(recPort);
			dp=new DatagramPacket(buf,buf.length);
			ds.receive(dp);
			ds.close();
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
		msgStr=new String(dp.getData(),0,dp.getLength());
		System.out.println("String" +msgStr);
		if(msgStr.equalsIgnoreCase("VOTE_COMMIT"))
		{
			scounter++;
		}
		System.out.println("Counter value="+scounter+"n Send value"+ssend);
	}
}
```

TPCClient1.java

```
import java.io.*;
import java.net.*;
public class TPCClient1
{
        public static void main(String a[]) throws Exception
        {
                InetAddress lclhost;
                lclhost=InetAddress.getLocalHost();
                Client clnt=new Client(lclhost);
                clnt.setSendPort(9001);
                clnt.setRecPort(8000);
                clnt.recData();
                clnt.sendData();
                clnt.recData();
        }
}
class Client
{
        InetAddress lclhost;
        int sendPort, recPort;
        Client(InetAddress lclhost)
        {
                this.lclhost=lclhost;
        }
        public void setSendPort(int sendPort)
        {
                this.sendPort=sendPort;
        }
        public void setRecPort(int recPort)
        {
                this.recPort=recPort;
        }
        public void sendData() throws Exception
```

```
	{
		BufferedReader br;
		DatagramSocket ds;
		DatagramPacket dp;
		String data="";
		System.out.println("Enter the response 'VOTE COMMIT'||'VOTE_ABORT'");
		br=new BufferedReader(new InputStreamReader(System.in));
		data=br.readLine();
		System.out.println("Data is"+data);
		ds=new DatagramSocket(sendPort);
		dp=new DatagramPacket(data.getBytes(),data.length(),lclhost,sendPort-1000);
		ds.send(dp);
		ds.close();
	}
	public void recData() throws Exception
	{
		byte buf[]=new byte[256];
		DatagramPacket dp;
		DatagramSocket ds;
		ds=new DatagramSocket(recPort);
		dp=new DatagramPacket(buf,buf.length);
		ds.receive(dp);
		ds.close();
		String msgStr=new String(dp.getData(),0,dp.getLength());
		System.out.println("Client1 Data"+msgStr);
	}
}
```

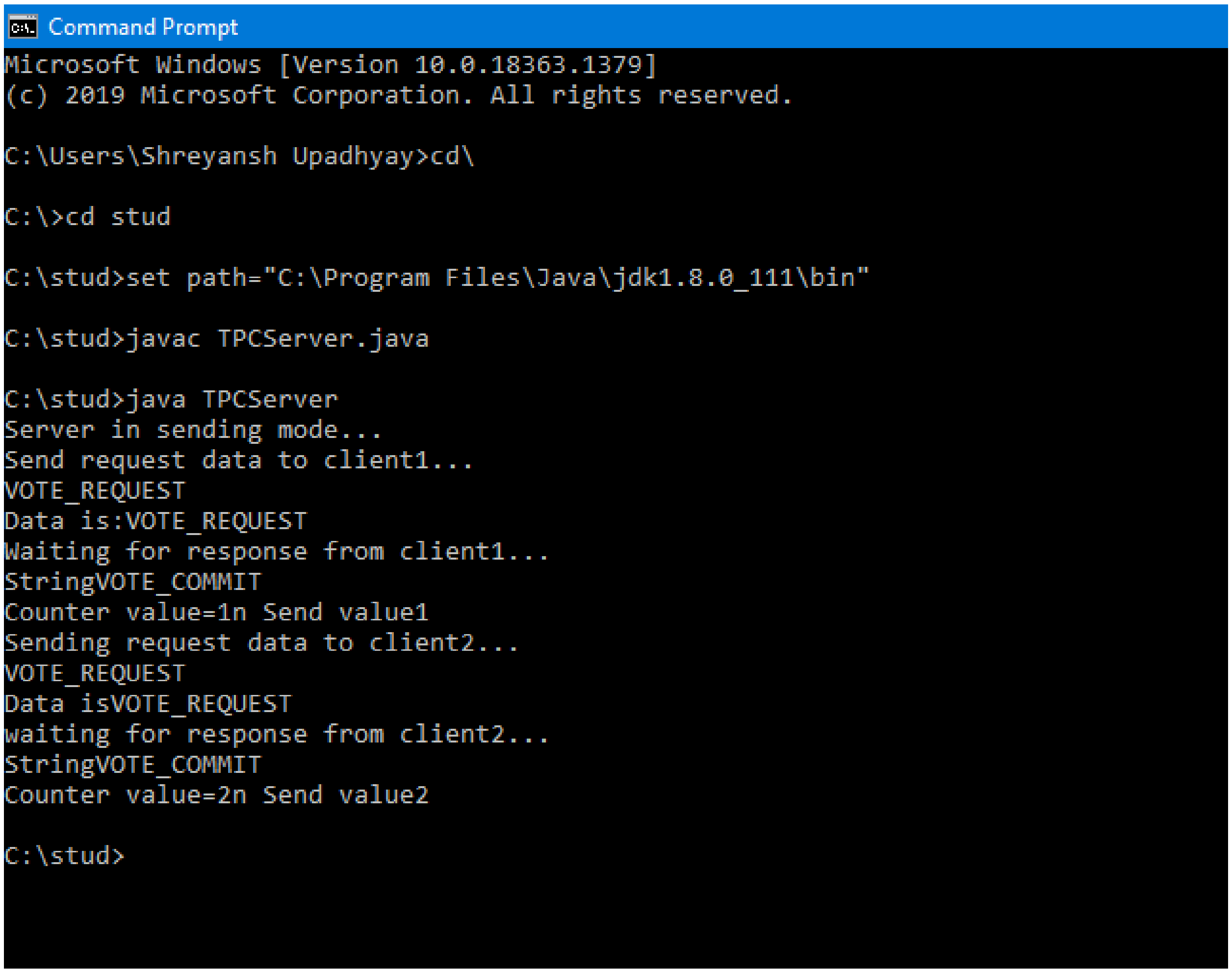
TPCClient2.java

```
import java.io.*;
import java.net.*;
public class TPCClient2
{
	public static void main(String a[]) throws Exception
	{
		InetAddress lclhost;
		lclhost=InetAddress.getLocalHost();
		Client client=new Client(lclhost);
		client.setSendPort(9003);
		client.setRecPort(8002);
```

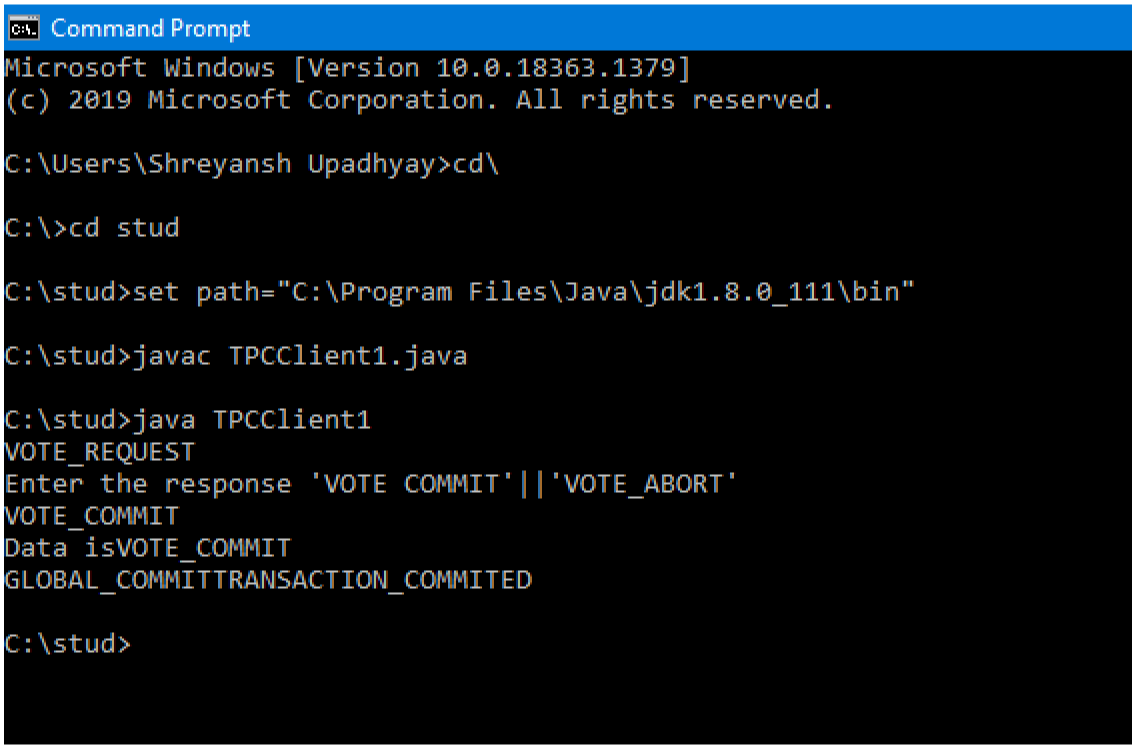
```
                client.recData();
                client.sendData();
                client.recData();
        }
}
class Client
{
        InetAddress lclhost;
        int sendPort, recPort;
        Client(InetAddress lclhost)
        {
                this.lclhost=lclhost;
        }
        public void setSendPort(int sendPort)
        {
                this.sendPort=sendPort;
        }
        public void setRecPort(int recPort)
        {
                this.recPort=recPort;
        }
        public void sendData() throws Exception
        {
                BufferedReader br;
                DatagramSocket ds;
                DatagramPacket dp;
                String data="";
                System.out.println("Enter the response 'VOTE COMMIT'||'VOTE_ABORT'");
                br=new BufferedReader(new InputStreamReader(System.in));
                data=br.readLine();
                System.out.println("Data is"+data);
                ds=new DatagramSocket(sendPort);
                dp=new DatagramPacket(data.getBytes(),data.length(),lclhost,sendPort-1000);
                ds.send(dp);
                ds.close();
        }
        public void recData() throws Exception
        {
                byte buf[]=new byte[256];
                DatagramPacket dp;
```

```
                    DatagramSocket ds;
                    ds=new DatagramSocket(recPort);
                    dp=new DatagramPacket(buf,buf.length);
                    ds.receive(dp);
                    ds.close();
                    String msgStr=new String(dp.getData(),0,dp.getLength());
                    System.out.println(msgStr);
          }
}
```

Output

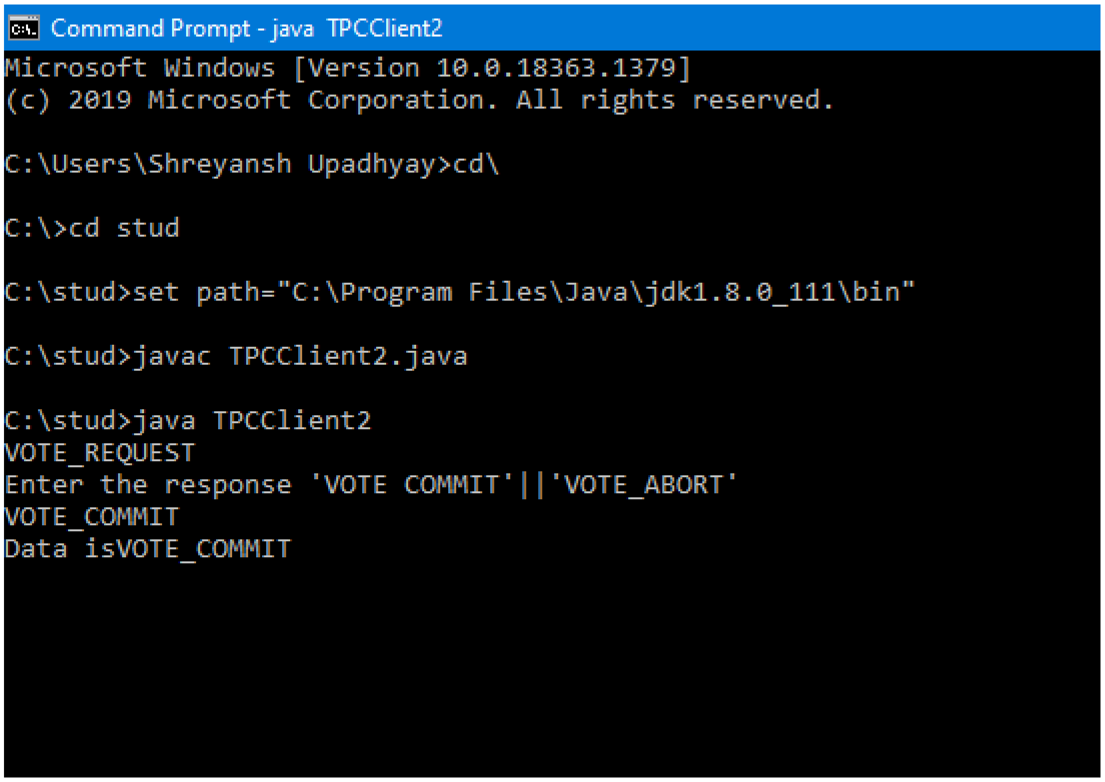


```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TPCServer.java

C:\stud>java TPCServer
Server in sending mode...
Send request data to client1...
VOTE_REQUEST
Data is:VOTE_REQUEST
Waiting for response from client1...
StringVOTE_COMMIT
Counter value=1n Send value1
Sending request data to client2...
VOTE_REQUEST
Data isVOTE_REQUEST
waiting for response from client2...
StringVOTE_COMMIT
Counter value=2n Send value2

C:\stud>
```



Command Prompt

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TPCClient1.java

C:\stud>java TPCClient1
VOTE_REQUEST
Enter the response 'VOTE COMMIT'||'VOTE_ABORT'
VOTE_COMMIT
Data isVOTE_COMMIT
GLOBAL_COMMITTRANSACTION_COMMITED

C:\stud>
```



Command Prompt - java TPCClient2

```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shreyansh Upadhyay>cd\

C:\>cd stud

C:\stud>set path="C:\Program Files\Java\jdk1.8.0_111\bin"

C:\stud>javac TPCClient2.java

C:\stud>java TPCClient2
VOTE_REQUEST
Enter the response 'VOTE COMMIT'||'VOTE_ABORT'
VOTE_COMMIT
Data isVOTE_COMMIT
```