

# **M.Sc. I.T. Part I**

## **Semester I**

### **DISTRIBUTED SYSTEM**

### **PRACTICAL 2018 – 2019**



**Vidyalankar School of Information Technology**  
**Wadala (East), Mumbai – 400037**

# **CERTIFICATE**

This is to certify that, **Mr. MD Taz Saheb** Seat Number \_\_\_\_\_  
studying in Master of Science in Information Technology Part-I (Semester-I)  
has satisfactorily completed the **Practical** in the subject of “**DISTRIBUTED  
SYSTEM**” as prescribed by University of Mumbai, during the academic year  
2018-2019.

**Subject Faculty**

**Coordinator, M.Sc. (IT)**

**External Examiner**

# INDEX

| <b><u>Sr.<br/>No.</u></b> | <b><u>Title</u></b>   | <b><u>Date</u></b> | <b><u>Page<br/>No.</u></b> | <b><u>Signature</u></b> |
|---------------------------|---|--------------------|----------------------------|-------------------------|
| 1.                        | Implement the concept for sharing the resources using distributed system. |                    |                            |                         |
| 2.                        | Write a program for implementing Client Server communication model.       |                    |                            |                         |
| 3.                        | Write a program to show the object communication using RMI.               |                    |                            |                         |
| 4.                        | Show the implementation of Remote Procedure Call.                         |                    |                            |                         |
| 5.                        | Show the implementation of Web Services.                                  |                    |                            |                         |
| 6.                        | Write a program to execute any one mutual exclusion algorithm.            |                    |                            |                         |
| 7.                        | Write a program to implement any one election algorithm.                  |                    |                            |                         |
| 8.                        | Show the implementation of any one clock synchronization algorithm.       |                    |                            |                         |
| 9.                        | Implement the concept of distributed file system architecture.            |                    |                            |                         |

[illegible]

## **Practical No: 01**

**Aim:** Implement the concept for sharing the resources using distributed system.

**Create a workgroup myworkgroup on windows machine and share a folder.**

If samba package is not installed on your linux machine then install it first.

```
#vi /etc/samba/smb.conf
```

Modify the file accordingly

```
Workgroup = MYGroup
```

```
Server String = Samba Server Version %v
```

```
Netbios name = MYSERVER
```

```
interface = lo eth0 192.154.2.10/24
```

```
host allow 127. 192.154.12.
```

**Remove the semicolon at the start of the above lines if any.**

**Go to end and add**

```
[common]
```

```
Comment=shared folder
```

```
Path = /common
```

```
Public =yes
```

```
Writable = yes
```

```
Browsable yes
```

```
Save the file
```

```
#useradd mona
```

```
#smbpasswd -a mona
```

```
Will ask for password
```

```
Enter the password
```

```
Retype the password
```

```
#/etc/init.d/smb restart
```

```
#chkconfig smb on
```

```
#testparm
```

```
#setenforce 0(zero)
```

```
#setsebool samba_enable_home+dirs=1
```

```
#getsebool -a | less
```

The ip address in the following command is the ip address of windows machine.

```
#smbclient //192.154.2.10/common -U mona
```

```
Enter password
```

```
Smb>mkdir ppbrdr
```

```
Will create ppbrdr directory in shared folder
```

```
Smb>get rdr.txt
```

```
Read the file rdr.txt
```

```
Smb>put ppb.txt
```

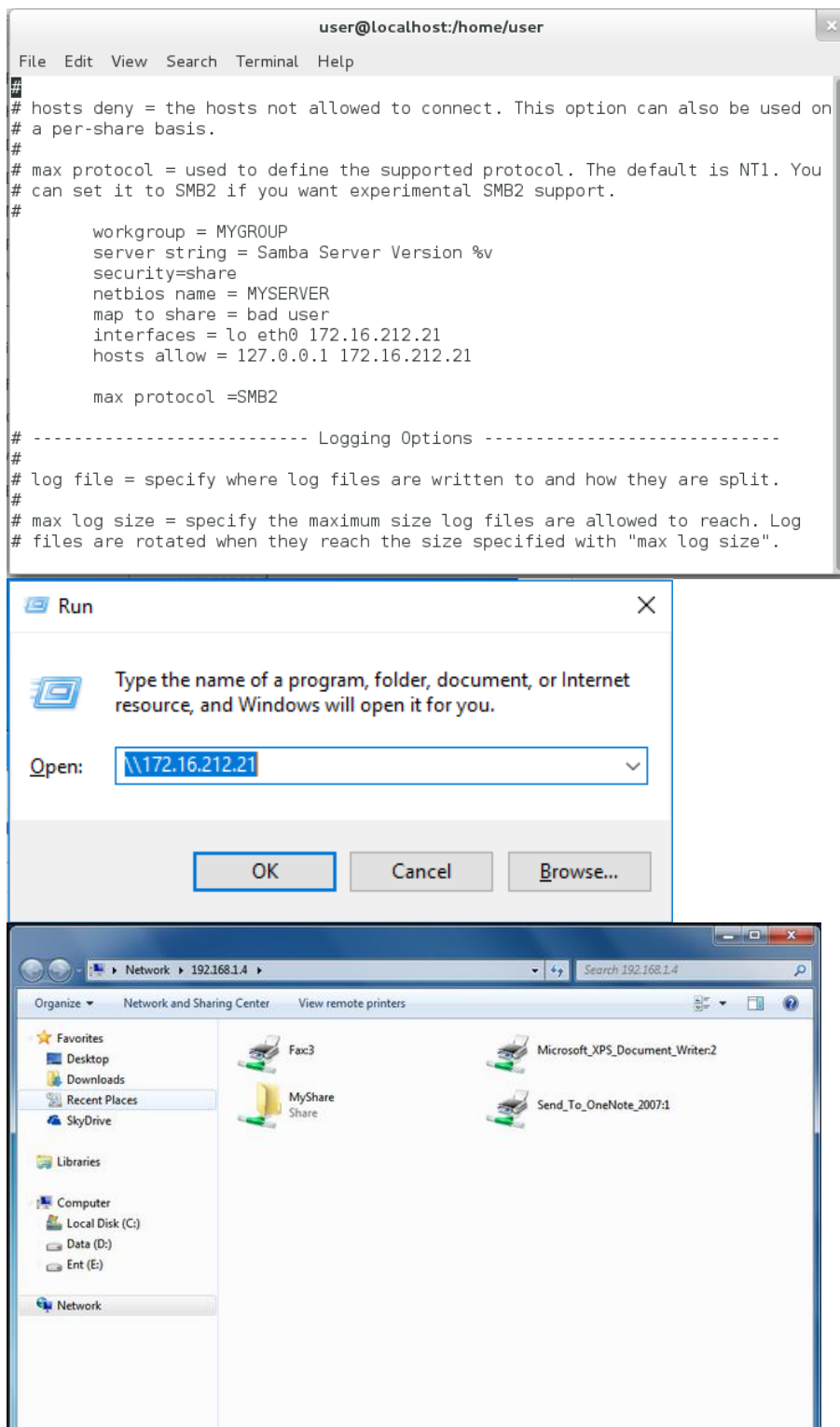
```
Will create a file named ppb.txt
```

```
user@localhost:/home/user
File Edit View Search Terminal Help
[user@localhost ~]$ su
Password:
[root@localhost user]# yum -y install samba samba-client samba-common
Loaded plugins: langpacks, presto, refresh-packagekit
Existing lock /var/run/yum.pid: another copy is running as pid 1997.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: PackageKit
    Memory : 222 M RSS (628 MB VSZ)
    Started: Thu Dec 20 10:21:08 2018 - 08:20 ago
    State : Sleeping, pid: 1997
Package 2:samba-4.0.11-1.fc18.x86_64 already installed and latest version
Package 2:samba-client-4.0.11-1.fc18.x86_64 already installed and latest version
Package 2:samba-common-4.0.11-1.fc18.x86_64 already installed and latest version
Nothing to do
[root@localhost user]# systemctl enable smb.service
ln -s '/usr/lib/systemd/system/smb.service' '/etc/systemd/system/multi-user.target.wants/smb.service'
[root@localhost user]# systemctl enable nmb.service
ln -s '/usr/lib/systemd/system/nmb.service' '/etc/systemd/system/multi-user.target.wants/nmb.service'
[root@localhost user]# systemctl start smb.service
[root@localhost user]# systemctl start nmb.service
[root@localhost user]# vi /etc/selinux/config
[root@localhost user]# firewall-cmd -add-service=samba
```

```
user@localhost:/home/user
File Edit View Search Terminal Help
TX packets 1167855 bytes 70567012 (67.2 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 16436
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 4 bytes 420 (420.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 420 (420.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost user]# vi /etc/selinux/config
[root@localhost user]# vi /etc/samba/smb.conf
[root@localhost user]# systemctl restart smb.service
[root@localhost user]# systemctl restart nmb.service
[root@localhost user]# firewall-cmd --add-service=samba
Warning: ALREADY_ENABLED
[root@localhost user]# vi /etc/samba/smb.conf
[root@localhost user]# systemctl restart smb.service
[root@localhost user]# systemctl restart nmb.service
[root@localhost user]# /etc/init.d/smb restart
bash: /etc/init.d/smb: No such file or directory
[root@localhost user]#
```



**Conclusion:** Resources are shared between Windows and Linux.

[illegible]



**Practical No: 02**

**Aim:** Write a program for implementing Client Server communication model.

A) A client server based program using TCP

**TCPClient.java**

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

class TCPClient
{
    public static void main(String args[ ]) throws Exception
    {
        Socket s = new Socket( "127.0.0.1", 2010 );
        Scanner kbin = new Scanner( System.in );
        DataInputStream in = new DataInputStream(s.getInputStream( ));
        DataOutputStream out = new DataOutputStream(s.getOutputStream( ));

        while( true )
        {
            out.writeUTF( kbin.next    ( ) );
            System.out.println( "Server: " +in.readUTF( ) );
        }

    }
}
```

**TCPServer.java**

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

class TCPServer
{
    public static void main(String args[ ]) throws Exception
    {
        ServerSocket ss = new ServerSocket( 2010 );

        while( true )
        {
            Socket s = ss.accept( );

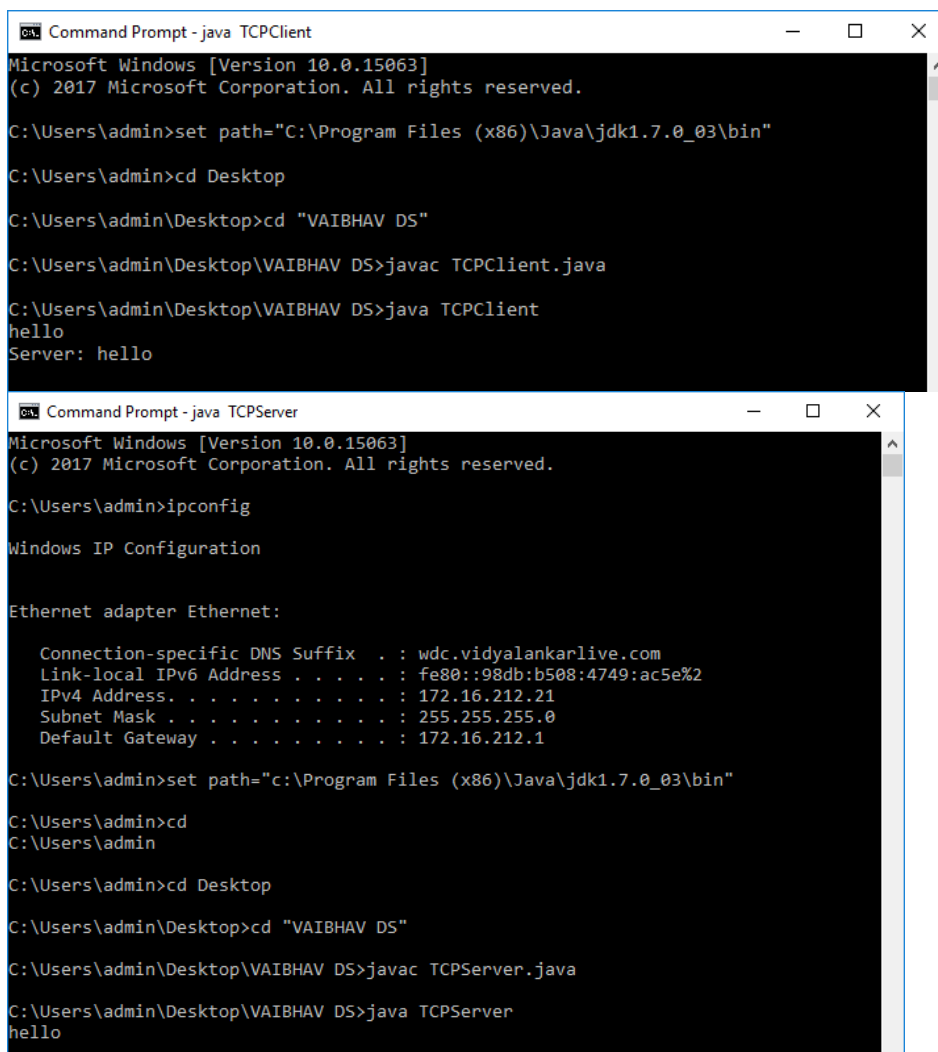
            DataInputStream in = new DataInputStream(s.getInputStream( ));
            DataOutputStream out = new DataOutputStream(s.getOutputStream( ));
            String msg="";

```

```
do
{
    msg = in.readUTF( );
    System.out.println(msg);
    out.writeUTF( msg );

}while( ! msg.equals("quit") );
}

}
```

**Output:**

The image shows two screenshots of Windows Command Prompts. The top screenshot is titled "Command Prompt - java TCPClient" and shows the following commands and output:

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\admin>set path="C:\Program Files (x86)\Java\jdk1.7.0_03\bin"

C:\Users\admin>cd Desktop

C:\Users\admin\Desktop>cd "VAIBHAV DS"

C:\Users\admin\Desktop\VAIBHAV DS>javac TCPClient.java

C:\Users\admin\Desktop\VAIBHAV DS>java TCPClient
hello
Server: hello
```

The bottom screenshot is titled "Command Prompt - java TCPServer" and shows the following commands and output:

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\admin>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : wdc.vidyalankarlive.com
    Link-local IPv6 Address . . . . . : fe80::98db:b508:4749:ac5e%2
    IPv4 Address. . . . . : 172.16.212.21
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 172.16.212.1

C:\Users\admin>set path="c:\Program Files (x86)\Java\jdk1.7.0_03\bin"

C:\Users\admin>cd
C:\Users\admin>

C:\Users\admin>cd Desktop

C:\Users\admin\Desktop>cd "VAIBHAV DS"

C:\Users\admin\Desktop\VAIBHAV DS>javac TCPServer.java

C:\Users\admin\Desktop\VAIBHAV DS>java TCPServer
hello
```

B) A client server based program for TCP marshalling of objects.

**Complex.java**

```
import java.io.*;

class Complex implements Serializable
{
    int a , b;

    Complex( int x, int y )
    {
        a = x;
        b = y;
    }

    public String toString( )
    {
        return "[ " + a + " + i " + b + " ]";
    }
}
```

**MarshallingServer.java**

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

class MarshallingServer
{
    public static void main(String args[ ] ) throws Exception
    {
        ServerSocket ss = new ServerSocket( 2010 );

        while( true )
        {
            Socket s = ss.accept( );
            System.out.println("Connection established...");

            ObjectOutputStream out = new ObjectOutputStream(s.getOutputStream( ));
            ObjectInputStream in = new ObjectInputStream(s.getInputStream( ));

            Complex c1 = (Complex) in.readObject();
            System.out.println( "Received object " +c1 );

            Complex c2 = (Complex) in.readObject();
            System.out.println( "Received object " +c2 );
        }
    }
}
```

```
        Complex c3 = new Complex( c1.a+c2.a , c1.b+c2.b );

        System.out.println( "Sending object " +c3 );
        out.writeObject( c3 );
        out.flush();

        s.close();
    }

}

}
```

**MarshallingClient.java**

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

class MarshallingClient
{
    public static void main(String args[ ] ) throws Exception
    {
        Socket s = new Socket( "127.0.0.1", 2010 );
        System.out.println( "connected");

        ObjectOutputStream out = new ObjectOutputStream(s.getOutputStream( ));
        ObjectInputStream in = new ObjectInputStream(s.getInputStream( ));

        Complex c1 = new Complex( 3, 5 );
        Complex c2 = new Complex( -1, 4 );

        out.writeObject( c1 );
        out.flush();
        out.writeObject( c2 );
        out.flush();
        System.out.println( "Sum is : " + (Complex) in.readObject( ) );

    }

}
```

**Output:**

```

Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\admin>set path="C:\Program Files (x86)\Java\jdk1.7.0_03\bin"

C:\Users\admin>cd Desktop

C:\Users\admin\Desktop>cd "VAIBHAV DS"

C:\Users\admin\Desktop\VAIBHAV DS>javac TCPClient.java

C:\Users\admin\Desktop\VAIBHAV DS>java TCPClient
hello
Server: hello
Exception in thread "main" java.util.NoSuchElementException

C:\Users\admin\Desktop\VAIBHAV DS>javac MarshallingClient.java

C:\Users\admin\Desktop\VAIBHAV DS>java MarshallingClient
connected
Sum is : [ 2 + i 9 ]

```

```

Select Command Prompt - java MarshallingServer
at TCPServer.main(TCPServer.java:19)

C:\Users\admin\Desktop\VAIBHAV DS>javac Complex.java

C:\Users\admin\Desktop\VAIBHAV DS>java Complex
Error: Main method not found in class Complex, please define the main method as:
    public static void main(String[] args)

C:\Users\admin\Desktop\VAIBHAV DS>javac MarshallingServer.java

C:\Users\admin\Desktop\VAIBHAV DS>java MarshallingServer
Connection established...
Received object [ 3 + i 5 ]
Received object [ -1 + i 4 ]
Sending object [ 2 + i 9 ]

```

C) A client server based program using UDP

**UDPSender.java**

```

import java.net.*;
class UDPSender
{
    public static void main(String args[]) throws Exception
    {
        InetAddress ia = InetAddress.getByName( "vsit-y101-07" );
        int port = 1099;
        String msg = " This is a great subject";
        byte []buf = msg.getBytes();
    }
}

```

```
DatagramPacket dp = new DatagramPacket( buf , buf.length, ia , port );

DatagramSocket ds = new DatagramSocket( ) ;

ds.send( dp );

System.out.println ( msg + " sent to " + ia + ":" +port );

    ds.receive( dp );

    String msg2 = new String( dp.getData() );

    System.out.println ("Echo Received "+msg2.trim());

}

}
```

**UDPReceiver.java**

```
import java.net.*;
import java.text.SimpleDateFormat;
import java.util.Date;

class UDPReceiver {
    public static void main(String args[]) throws Exception
    {
        DatagramSocket ds = new DatagramSocket( 1099 ) ;
        while( true ){
            byte []buf = new byte [1000];

            DatagramPacket dp = new DatagramPacket( buf , buf.length );

            ds.receive( dp );

            String msg = new String( dp.getData() );

            Date date = new Date();

            System.out.println ( "Received : " + msg.trim() + " from " +
dp.getAddress() + ":" + dp.getPort() + " at " + date );

            ds.send(dp);

        }

    }

}
```

**Output:**

```
C:\Users\admin\Desktop\VAIBHAV DS>javac UDPReceiver.java

C:\Users\admin\Desktop\VAIBHAV DS>java UDPReceiver
Received : This is a great subject from /172.16.212.21:53467 at Thu Dec 20 12:44:55
IST 2018
```

```
C:\Users\admin\Desktop\VAIBHAV DS>javac UDPSender.java

C:\Users\admin\Desktop\VAIBHAV DS>java UDPSender
This is a great subject sent to vsit-y101-07/172.16.212.21:1099
Echo Received This is a great subject

C:\Users\admin\Desktop\VAIBHAV DS>
```

D) A client server based program for UDP echo chat.

**UDPCClient.java**

```
import java.net.*;
class UDPSender {
    public static void main(String args[]) throws Exception
    {
        InetAddress ia = InetAddress.getByName( "LAPTOP-R8KE5PAE" );
        int port = 1099;
        String msg = " This is Aditya";
        byte []buf = msg.getBytes();

        DatagramPacket dp = new DatagramPacket( buf , buf.length, ia , port );
        DatagramSocket ds = new DatagramSocket( ) ;
        ds.send( dp );
        System.out.println ( msg + " sent to " + ia + ":" +port );
        ds.receive( dp );
        String msg2 = new String( dp.getData() );
        System.out.println ("Echo Received "+msg2.trim());
    }
}
```

**UDPServer.java**

```
import java.net.*;
import java.text.SimpleDateFormat;
import java.util.Date;

class UDPReceiver {

    public static void main(String args[]) throws Exception
    {
        DatagramSocket ds = new DatagramSocket( 1099 ) ;

        while( true ){
            byte []buf = new byte [1000];

            DatagramPacket dp = new DatagramPacket( buf , buf.length );
            ds.receive( dp );

            String msg = new String( dp.getData() );

            Date date = new Date();

            System.out.println ( "Received : " + msg.trim() + " from " +
dp.getAddress() + ":" + dp.getPort() + " at " + date );

            ds.send(dp);
        }
    }
}
```



**Output:**

```
Command Prompt
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Aditya>cd desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd echo
C:\Users\Aditya\Desktop\DS\echo>set path="C:\Program Files (x86)\Java\jdk1.6.0_02\bin"
C:\Users\Aditya\Desktop\DS\echo>javac UDPSender.java
C:\Users\Aditya\Desktop\DS\echo>java UDPSender
This is Aditya sent to LAPTOP-R8KE5PAE/192.168.32.1:1099
Echo Received This is Aditya
C:\Users\Aditya\Desktop\DS\echo>
```

```
Command Prompt - java UDPreceiver
-extendeddirs <dirs>          Override location of installed extensions
-endsorddirs <dirs>          Override location of endorsed standards path
-proc:{none,only}             Control whether annotation processing and/or compilation is done.
-processor <class1>[,<class2>,<class3>...]Names of the annotation processors to run; bypasses default discovery proces
-s
-processorpath <path>         Specify where to find annotation processors
-d <directory>               Specify where to place generated class files
-s <directory>               Specify where to place generated source files
-implicit:{none,class}       Specify whether or not to generate class files for implicitly referenced files
-encoding <encoding>         Specify character encoding used by source files
-source <release>             Provide source compatibility with specified release
-target <release>            Generate class files for specific VM version
-version                     Version information
-help                        Print a synopsis of standard options
-Akey[=value]                Options to pass to annotation processors
-X                            Print a synopsis of nonstandard options
-J<flag>                     Pass <flag> directly to the runtime system

C:\Users\Aditya>cd Desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd echo
C:\Users\Aditya\Desktop\DS\echo>javac UDPreceiver.java
C:\Users\Aditya\Desktop\DS\echo>java UDPreceiver
Received : This is Aditya from /192.168.32.1:62236 at Mon Dec 17 23:51:23 IST 2018
```

E) A multicast Socket program

**MulticastSender.java**

```
import java.net.*;
class MulticastSender
{
    public static void main( String args[] )throws Exception
    {
        // Which port should we send to
        int port = 5000;
        // Which address
        String group = "225.4.5.6";
        // Create the socket but we don't bind it as we are only going to send data
        MulticastSocket s = new MulticastSocket();
        // Note that we don't have to join the multicast group if we are only
        // sending data and not receiving
        // Fill the buffer with some data
        byte buf[] = new java.util.Date().toString().getBytes();

        // Create a DatagramPacket
        DatagramPacket pack = new DatagramPacket(buf, buf.length,
                                                InetAddress.getByName(group), port);
        // Do a send. Note that send takes a byte for the ttl and not an int.
        s.send(pack);
        // And when we have finished sending data close the socket
        s.close();
    }
}
```

**MulticastReceiver.java**

```
import java.net.*;
class MulticastReceiver
{
    public static void main(String args[]) throws Exception
    {
        // Which port should we listen to
        int port = 5000;
        // Which address
        String group = "225.4.5.6";
        // Create the socket and bind it to port 'port'.
        MulticastSocket s = new MulticastSocket(port);
        // join the multicast group
        s.joinGroup(InetAddress.getByName(group));
        int ctr=0;
        while(true)
        {
```

```
// Now the socket is set up and we are ready to receive packets
// Create a DatagramPacket and do a receive
byte buf[] = new byte[1024];
DatagramPacket pack = new DatagramPacket(buf, buf.length);
s.receive(pack);
// Finally, let us do something useful with the data we just received,
// like print it on stdout :-
System.out.println("Received data from: " + pack.getAddress() +
    ":" + pack.getPort() + "\t" );
System.out.println(new String(pack.getData()).trim());
System.out.println();
// And when we have finished receiving data leave the multicast group
    if( ++ctr == 999) break;
}
// close the socket
s.leaveGroup(InetAddress.getByName(group));
s.close();
}
}
```

**Output:**

```
C:\Users\admin\Desktop\VAIBHAV DS>javac MulticastSender.java
C:\Users\admin\Desktop\VAIBHAV DS>java MulticastSender
C:\Users\admin\Desktop\VAIBHAV DS>java MulticastSender
C:\Users\admin\Desktop\VAIBHAV DS>java MulticastSender
C:\Users\admin\Desktop\VAIBHAV DS>
```

```
C:\Users\admin\Desktop\VAIBHAV DS>javac MulticastReceiver.java
C:\Users\admin\Desktop\VAIBHAV DS>java MulticastReceiver
Received data from: /172.16.212.21:63898
Thu Dec 20 12:59:14 IST 2018

Received data from: /172.16.212.21:63904
Thu Dec 20 12:59:39 IST 2018

Received data from: /172.16.212.21:63906
Thu Dec 20 12:59:41 IST 2018
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

### Practical No: 03

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

**Code:**

**Adder.java**

```
import java.rmi.*;
public interface Adder extends Remote{

    public int add(int x,int y)throws RemoteException;
}
```

**AdderRemote.java**

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

public class AdderRemote extends UnicastRemoteObject implements Adder
{

    public AdderRemote()throws RemoteException {
        super();
    }

    public int add(int x,int y) {
        return x+y;
    }

}
```

**MyClient.java**

```
import java.rmi.*;

public class MyClient{

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

            Adder x=(Adder)Naming.lookup("rmi://localhost:5555/vsitadderservice");
            System.out.println(x.add(34,4));

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

    }

}
```

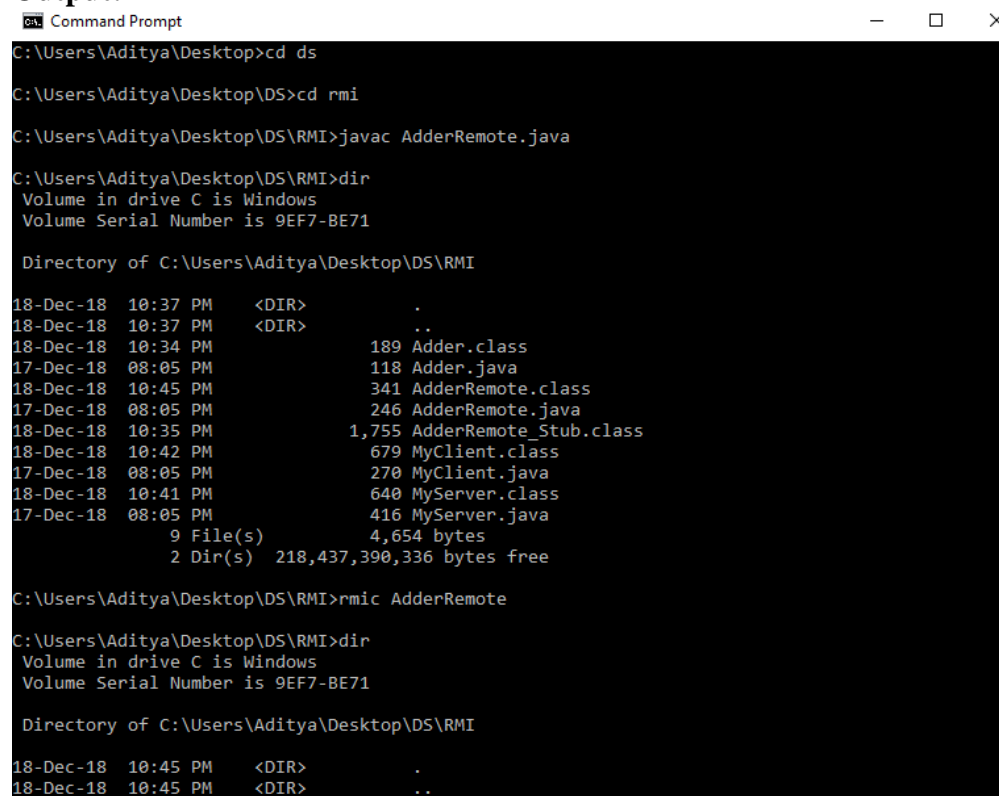
**MyServer.java**

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

public class MyServer{

    public static void main(String args[]){
        try{
            // variable will be of Interface i.e. a1 and Object will be created
            // of that class who has extended the interface i.e. AdderRemote()
            Adder a1=new AdderRemote();
            Naming.rebind("rmi://localhost:5555/vsitadderservice",a1);

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

**Output:**

```
Command Prompt
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd rmi
C:\Users\Aditya\Desktop\DS\RM1>javac AdderRemote.java
C:\Users\Aditya\Desktop\DS\RM1>dir
Volume in drive C is Windows
Volume Serial Number is 9EF7-BE71

Directory of C:\Users\Aditya\Desktop\DS\RM1

18-Dec-18 10:37 PM <DIR>      .
18-Dec-18 10:37 PM <DIR>      ..
18-Dec-18 10:34 PM          189 Adder.class
17-Dec-18 08:05 PM          118 Adder.java
18-Dec-18 10:45 PM          341 AdderRemote.class
17-Dec-18 08:05 PM          246 AdderRemote.java
18-Dec-18 10:35 PM       1,755 AdderRemote_Stub.class
18-Dec-18 10:42 PM          679 MyClient.class
17-Dec-18 08:05 PM          270 MyClient.java
18-Dec-18 10:41 PM          640 MyServer.class
17-Dec-18 08:05 PM          416 MyServer.java
                9 File(s)      4,654 bytes
                2 Dir(s)  218,437,390,336 bytes free

C:\Users\Aditya\Desktop\DS\RM1>rmic AdderRemote
C:\Users\Aditya\Desktop\DS\RM1>dir
Volume in drive C is Windows
Volume Serial Number is 9EF7-BE71

Directory of C:\Users\Aditya\Desktop\DS\RM1

18-Dec-18 10:45 PM <DIR>      .
18-Dec-18 10:45 PM <DIR>      ..
```

```

Command Prompt
17-Dec-18 08:05 PM 416 MyServer.java
9 File(s) 4,654 bytes
2 Dir(s) 218,437,390,336 bytes free

C:\Users\Aditya\Desktop\DS\RM1>rmic AdderRemote

C:\Users\Aditya\Desktop\DS\RM1>dir
Volume in drive C is Windows
Volume Serial Number is 9EF7-BE71

Directory of C:\Users\Aditya\Desktop\DS\RM1

18-Dec-18 10:45 PM <DIR> .
18-Dec-18 10:45 PM <DIR> ..
18-Dec-18 10:34 PM 189 Adder.class
17-Dec-18 08:05 PM 118 Adder.java
18-Dec-18 10:45 PM 341 AdderRemote.class
17-Dec-18 08:05 PM 246 AdderRemote.java
18-Dec-18 10:45 PM 1,755 AdderRemote_Stub.class
18-Dec-18 10:42 PM 679 MyClient.class
17-Dec-18 08:05 PM 270 MyClient.java
18-Dec-18 10:41 PM 640 MyServer.class
17-Dec-18 08:05 PM 416 MyServer.java
9 File(s) 4,654 bytes
2 Dir(s) 218,437,251,072 bytes free

C:\Users\Aditya\Desktop\DS\RM1>start rmiregistry 5555

C:\Users\Aditya\Desktop\DS\RM1>
    
```

```

Command Prompt - java MyServer

-implicit:<none,class> Specify whether or not to generate class file
s for implicitly referenced files
-encoding <encoding> Specify character encoding used by source fil
es
-source <release> Provide source compatibility with specified r
elease
-target <release> Generate class files for specific VM version
-version Version information
-help Print a synopsis of standard options
-Akey[=value] Options to pass to annotation processors
-X Print a synopsis of nonstandard options
-J<flag> Pass <flag> directly to the runtime system

C:\Users\Aditya>cd Desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd rmi
C:\Users\Aditya\Desktop\DS\RM1>javac MyServer.java
C:\Users\Aditya\Desktop\DS\RM1>java MyServer
    
```

```

Command Prompt

plicitly referenced files
-encoding <encoding> Specify character encoding used by source files
-source <release> Provide source compatibility with specified release
-target <release> Generate class files for specific VM version
-version Version information
-help Print a synopsis of standard options
-Akey[=value] Options to pass to annotation processors
-X Print a synopsis of nonstandard options
-J<flag> Pass <flag> directly to the runtime system

C:\Users\Aditya>cd Desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS> cd rmi
C:\Users\Aditya\Desktop\DS\RM1>javac MyClient.java
C:\Users\Aditya\Desktop\DS\RM1>java MyClient
38
C:\Users\Aditya\Desktop\DS\RM1>
    
```

[illegible]



**Practical No: 04**

**Aim:** Show the implementation of Remote Procedure Call.

**Code:**

**RPCClient.java**

```
import java.io.*;
import java.net.*;
class RPCClient
{
    public static void main(String[] args)
    {
        try
        {
            InetAddress ia = InetAddress.getLocalHost();
            DatagramSocket ds = new DatagramSocket();
            DatagramSocket ds1 = new DatagramSocket(1300);
            System.out.println("\nRPC Client\n");
            System.out.println("Enter method name and parameter (eg. add 3 4)\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();}
    }
}
```

**RPCServer.java**

```
import java.util.*;
import java.net.*;
class RPCServer
{
    DatagramSocket ds;
    DatagramPacket dp;

    RPCServer()
    {
```

```
String str="",methodName="";
int val1=0,val2=0;
double result=0;

try{
    ds=new DatagramSocket(1200);
    while(true)
    {
        byte b[]=new byte[1000];
        dp=new DatagramPacket(b,b.length);
        ds.receive(dp);

        str=new String(dp.getData(),0,dp.getLength());

        if(str.equalsIgnoreCase("q"))
            continue;

        StringTokenizer st = new StringTokenizer(str," ");
        int i=0;
        while(st.hasMoreTokens())
        {
            methodName = st.nextToken();
            val1 = Integer.parseInt(st.nextToken());
            val2 = Integer.parseInt(st.nextToken());
        }

        System.out.println(str);
        InetAddress ia = InetAddress.getLocalHost();

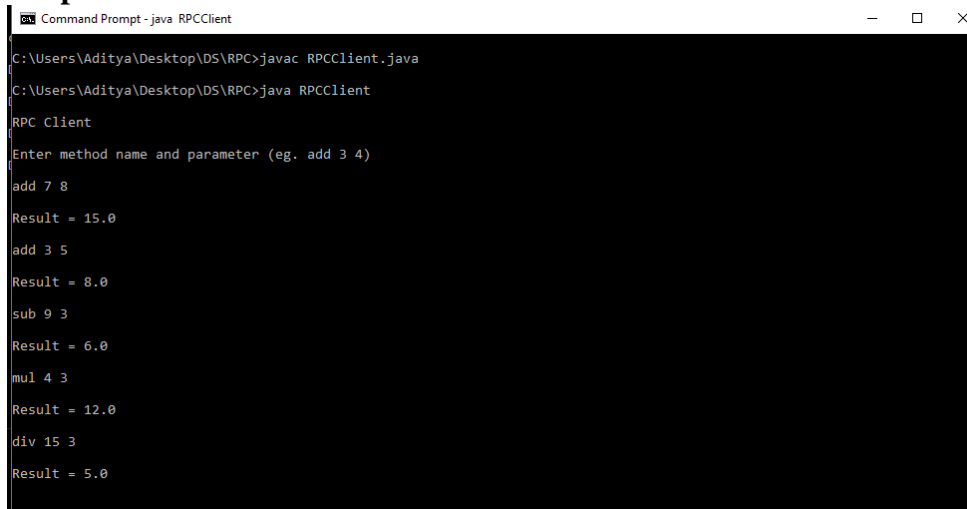
        if(methodName.equals("add"))    result = add(val1,val2);
        else if(methodName.equals("sub")) result = sub(val1,val2);
        else if(methodName.equals("mul")) result = mul(val1,val2);
        else if(methodName.equals("div")) result = div(val1,val2);

        byte b1[]=new String(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 add(int val1, int val2)
{
    return val1+val2;
}
```

```
public double sub(int val3, int val4)
{
    return val3-val4;
}
public double mul(int val3, int val4)
{
    return val3*val4;
}
public double div(int val3, int val4)
{
    return val3 / (double)val4;
}
public static void main(String[] args)
{
    new RPCServer();
}
}
```

**Output:**

```
Command Prompt - java RPCClient
C:\Users\Aditya\Desktop\DS\RPC>javac RPCClient.java
C:\Users\Aditya\Desktop\DS\RPC>java RPCClient
RPC Client
Enter method name and parameter (eg. add 3 4)
add 7 8
Result = 15.0
add 3 5
Result = 8.0
sub 9 3
Result = 6.0
mul 4 3
Result = 12.0
div 15 3
Result = 5.0
```

```
Command Prompt - java RPCServer
-Akey[=value]    Options to pass to annotation processors
-X              Print a synopsis of nonstandard options
-J<flag>         Pass <flag> directly to the runtime system

C:\Users\Aditya>cd desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd rpc
C:\Users\Aditya\Desktop\DS\RPC>javac RPCServer.java
C:\Users\Aditya\Desktop\DS\RPC>java RPCServer
add 7 8
result : 15.0

add 3 5
result : 8.0

sub 9 3
result : 6.0

mul 4 3
result : 12.0

div 15 3
result : 5.0
```

This image shows a full page of white paper with horizontal black ruling lines. The lines are evenly spaced and run across the width of the page, providing a template for writing or drawing. There are no margins, text, or other markings on the page.

## Practical No. 05

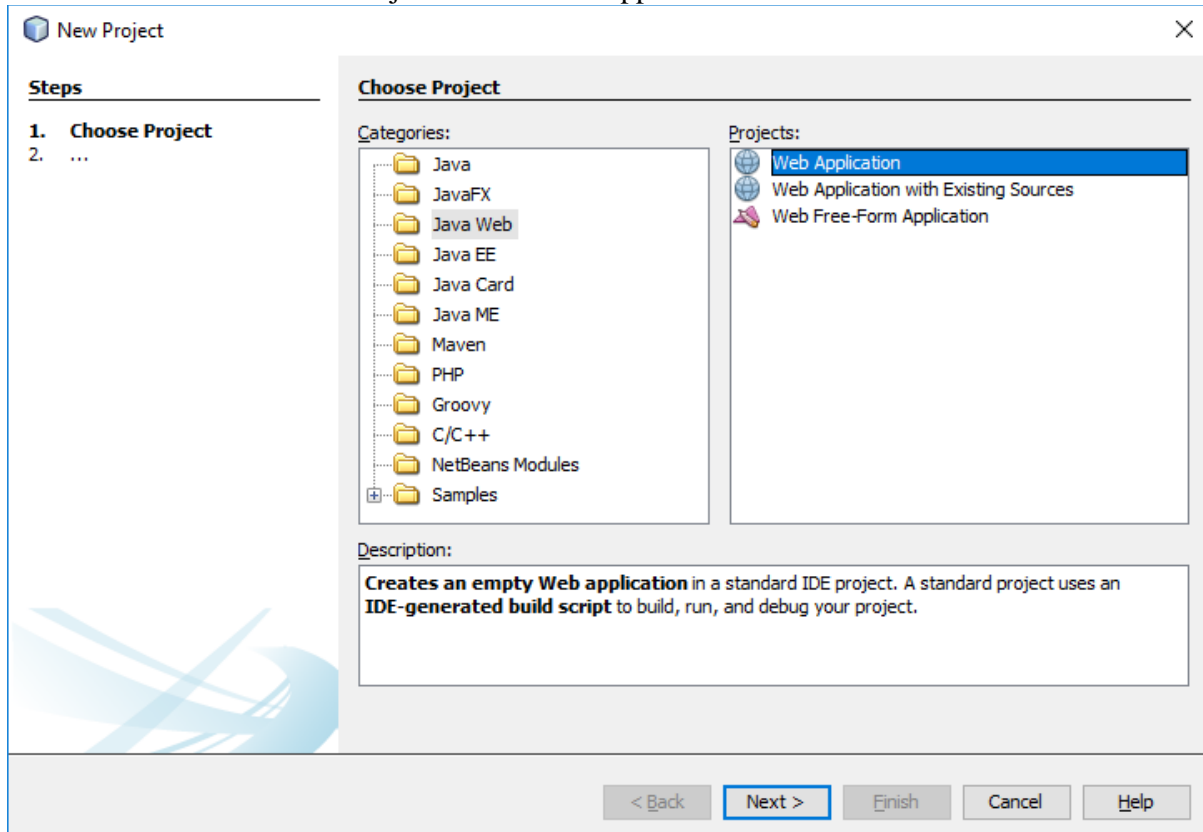
**Aim:** Show the implementation of web services.

**A:** Implementing “Big” Web Service.

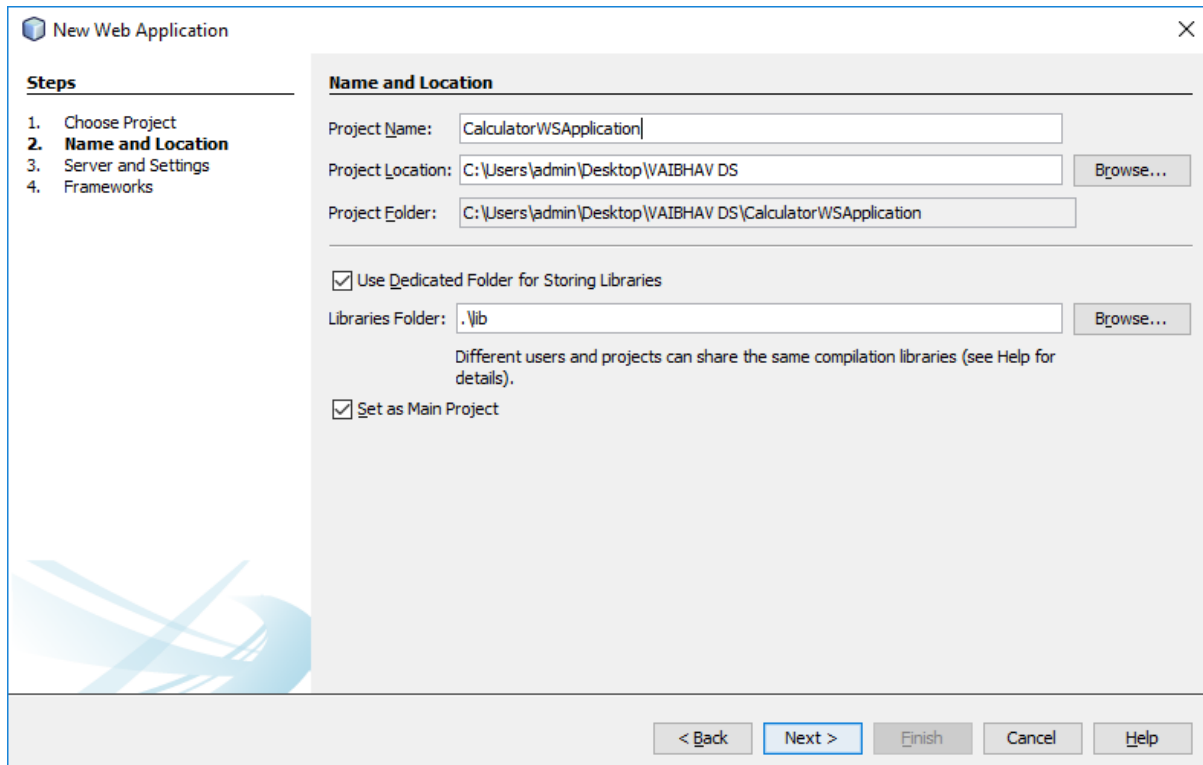
### 1) Creating a Web Service

#### A. Choosing a Container:

1. Choose File > New Project. Select Web Application from the Java Web.



2. Name the project CalculatorWSApplication. Select a location for the project. Click Next.



**New Web Application**

**Steps**

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

**Name and Location**

Project Name:

Project Location:

Project Folder:

☒ Use Dedicated Folder for Storing Libraries

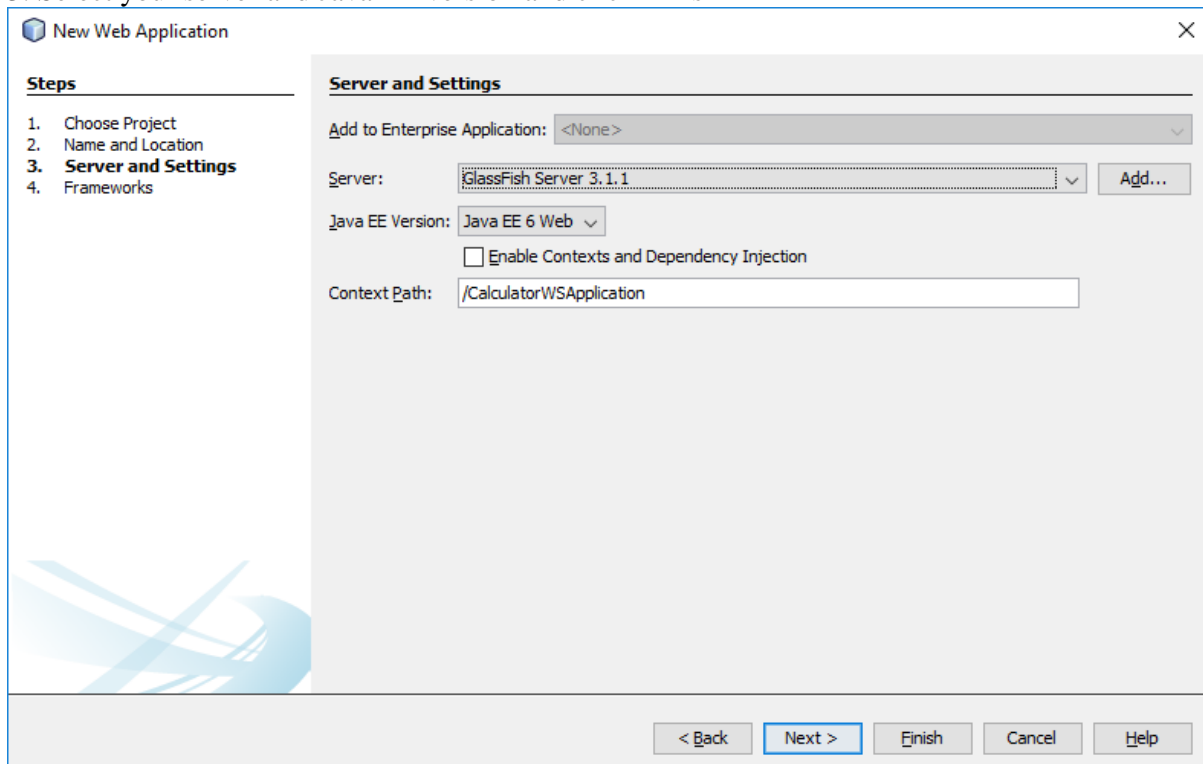
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☒ Set as Main Project

< Back   Next >   Finish   Cancel   Help

### 3. Select your server and Java EE version and click Finish



**New Web Application**

**Steps**

1. Choose Project
2. Name and Location
- 3. Server and Settings**
4. Frameworks

**Server and Settings**

Add to Enterprise Application:

Server:

Java EE Version:

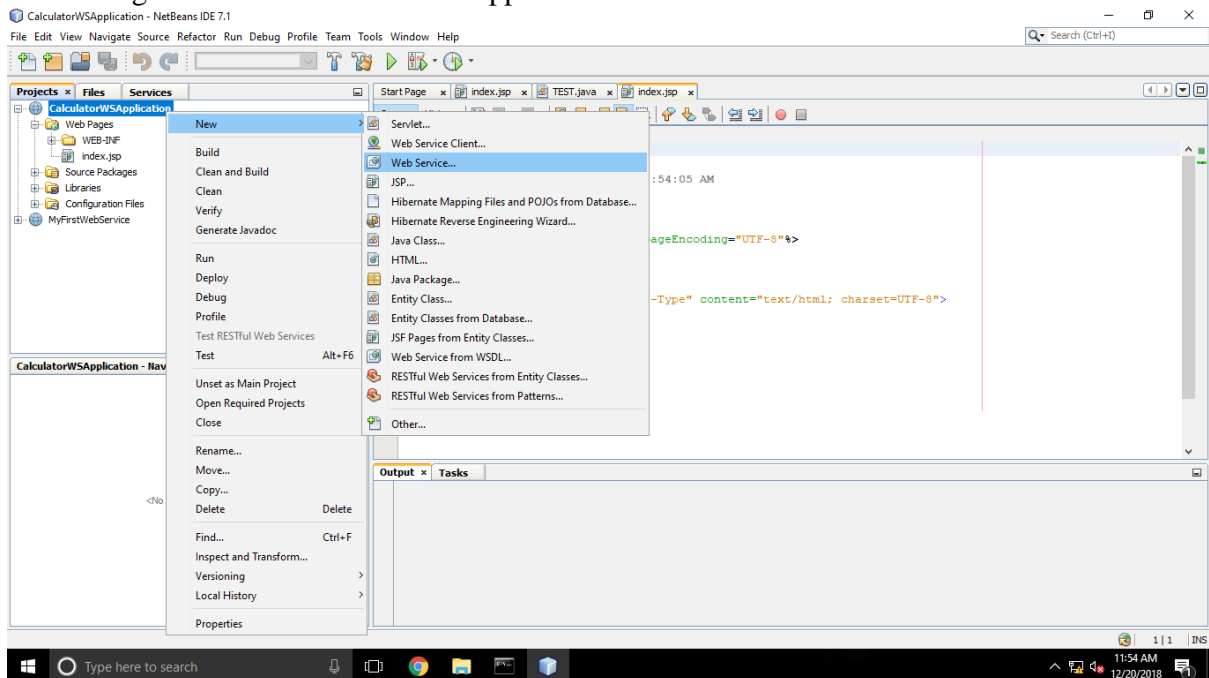
☐ Enable Contexts and Dependency Injection

Context Path:

< Back   Next >   Finish   Cancel   Help

## B. Creating a Web Service from a Java Class

1. Right-click the CalculatorWSApplication node and choose New > Web Service.



2. Name the web service CalculatorWS and type org.me.calculator in Package. Leave Create Web Service from Scratch selected. If you are creating a Java EE 6 project on GlassFish or WebLogic, select Implement Web Service as a Stateless Session Bean.



**New Web Service**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Web Service Name: CalculatorWS

Project: CalculatorWSApplication

Location: Source Packages

Package: org.me.calculator

☒ Create Web Service from Scratch

☐ Create Web Service from Existing Session Bean

Enterprise Bean:

☒ Implement Web Service as Stateless Session Bean

< Back   Next >   **Finish**   Cancel   Help

3. Click Finish. The Projects window displays the structure of the new web service and the source code is shown in the editor area.

## 2) Adding an Operation to the Web Service

The goal of this exercise is to add to the web service an operation that adds two numbers received from a client. The NetBeans IDE provides a dialog for adding an operation to a web service. You can open this dialog either in the web service visual designer or in the web service context menu.

### A. To add an operation to the web service:

1. Change to the Design view in the editor.
2. Click Add Operation in either the visual designer or the context menu. The Add Operation dialog opens.
3. In the upper part of the Add Operation dialog box, type add in Name and type int in the Return Type drop-down list.
4. In the lower part of the Add Operation dialog box, click Add and create a parameter of type int named i.
5. Click Add again and create a parameter of type int called j. You now see the following:

**Add Operation...**

Name:

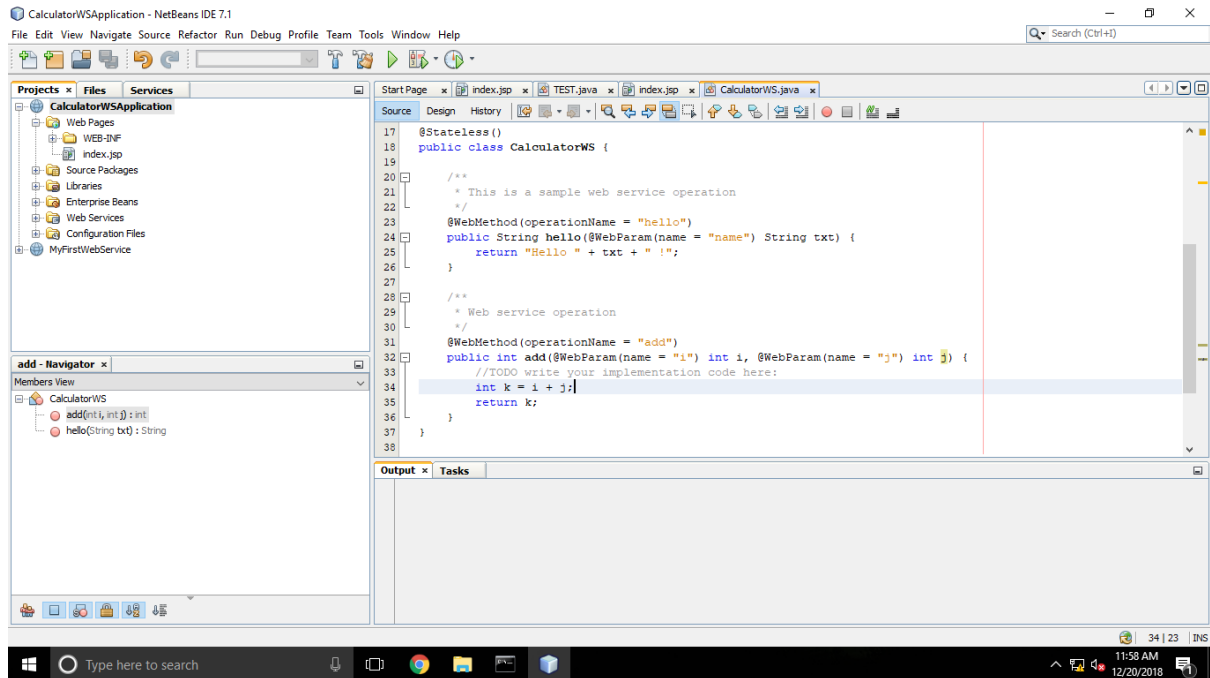
Return Type:

Parameters Exceptions

| Name | Type | Final                    |
|------|------|--------------------------|
| i    | int  | <input type="checkbox"/> |
| j    | int  | <input type="checkbox"/> |

6. Click OK at the bottom of the Add Operation dialog box. You return to the editor.
7. The visual designer now displays the following:
8. Click Source. And code the following.

```
@WebMethod(operationName = "add")
public int add(@WebParam(name = "i") int i, @WebParam(name = "j") int j)
{
    int k = i + j;
    return k;
}
```

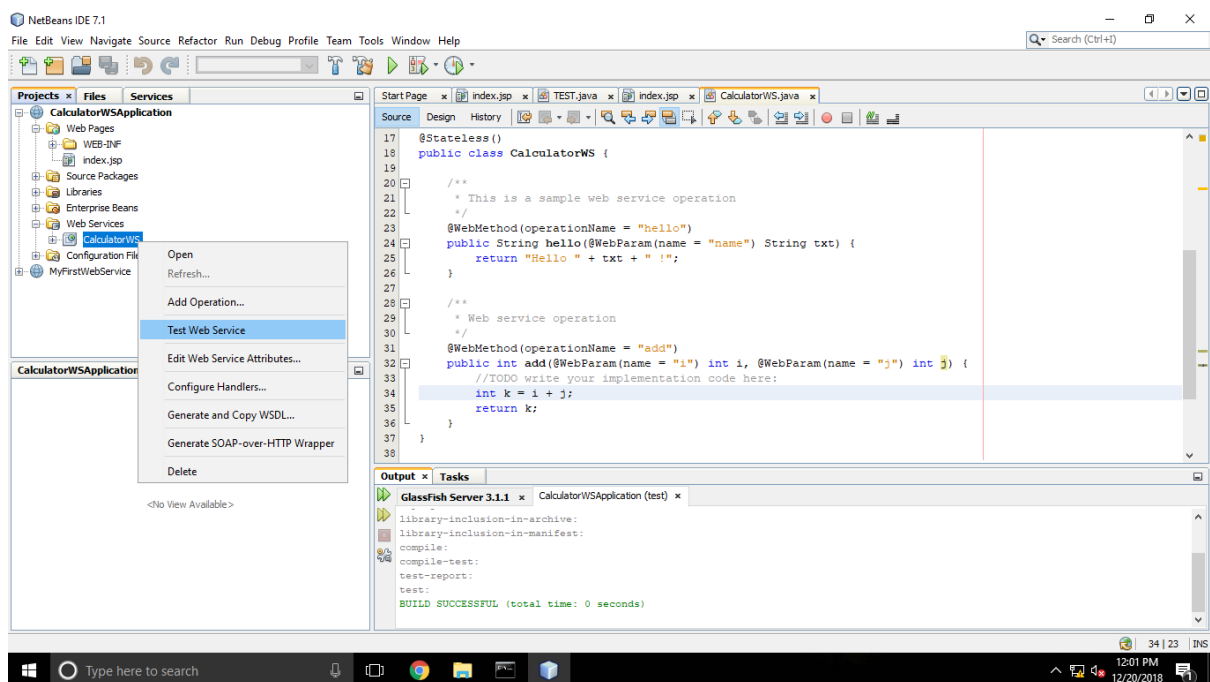


### 3) Deploying and Testing the Web Service

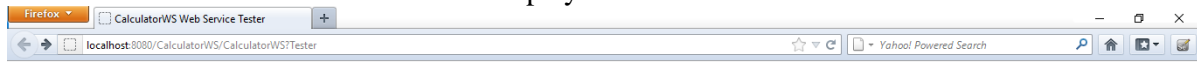
After you deploy a web service to a server, you can use the IDE to open the server's test client, if the server has a test client. The GlassFish and WebLogic servers provide test clients.

#### A. To test successful deployment to a GlassFish or WebLogic server:

1. Right-click the project and choose Deploy. The IDE starts the application server, builds the application, and deploys the application to the server.
2. In the IDE's Projects tab, expand the Web Services node of the CalculatorWSApplication project. Right-click the CalculatorWS node, and choose Test Web Service.



3. The IDE opens the tester page in your browser, if you deployed a web application to the GlassFish server.
4. If you deployed to the GlassFish server, type two numbers in the tester page, as shown below
5. The sum of the two numbers is displayed:



## CalculatorWS Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

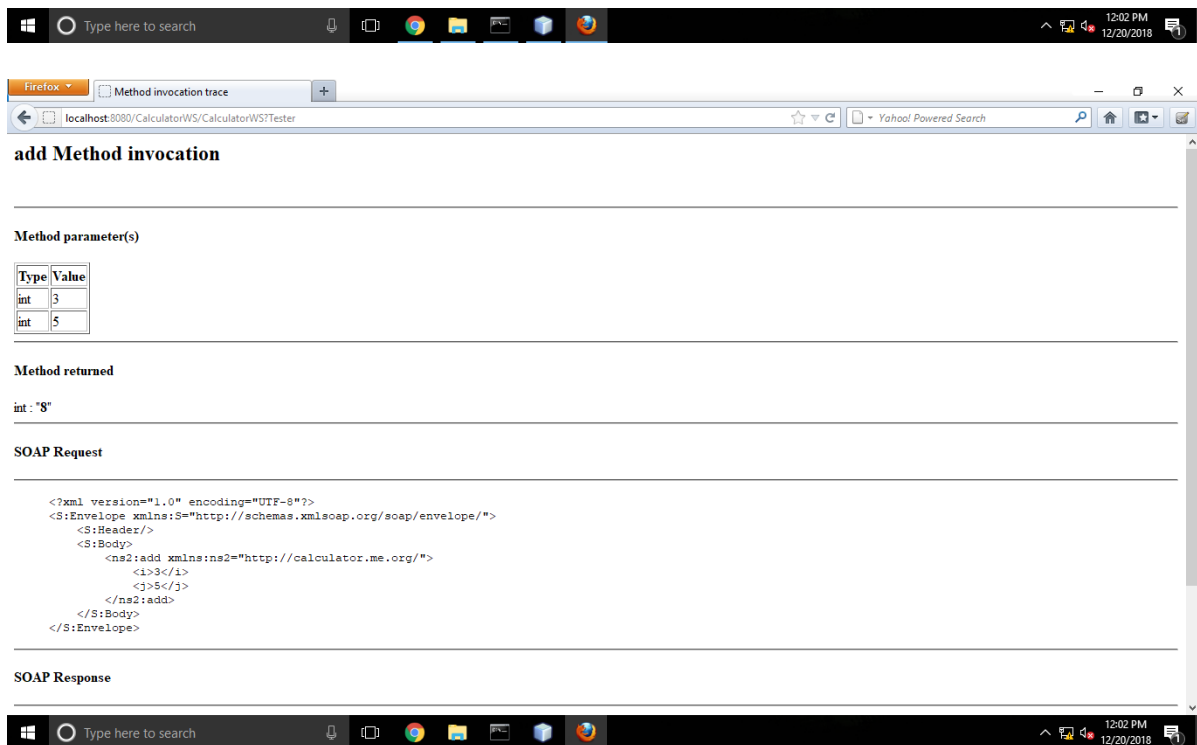
### Methods :

public abstract int org.me.calculator.CalculatorWS.add(int,int)

add (3 5)

public abstract java.lang.String org.me.calculator.CalculatorWS.hello(java.lang.String)

hello ( )

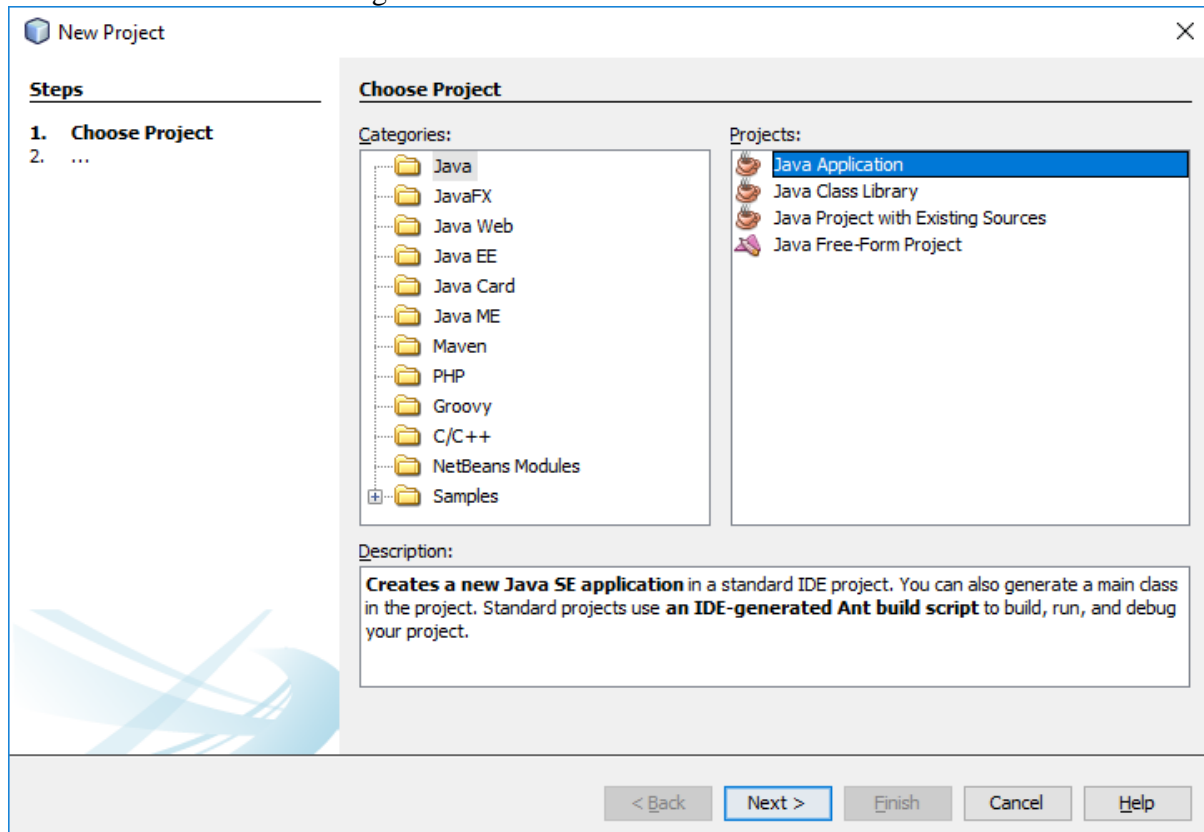


#### 4) Consuming the Web Service

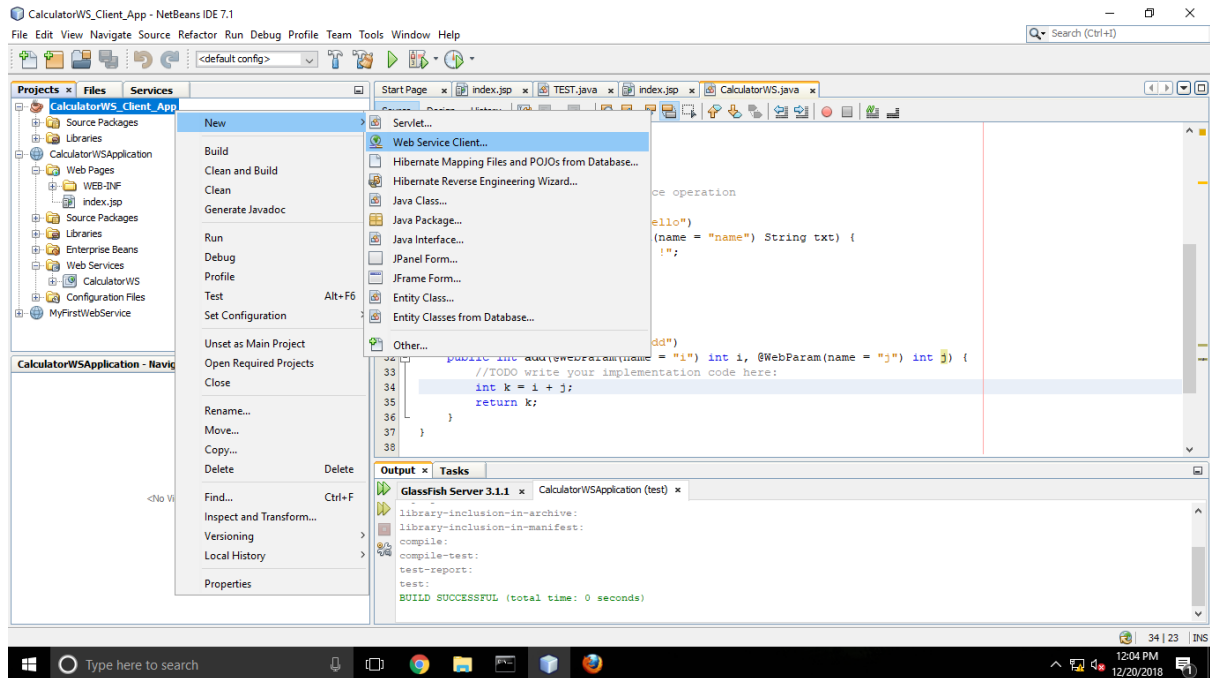
Now that you have deployed the web service, you need to create a client to make use of the web service's add method.

##### 1. Client: Java Class in Java SE Application

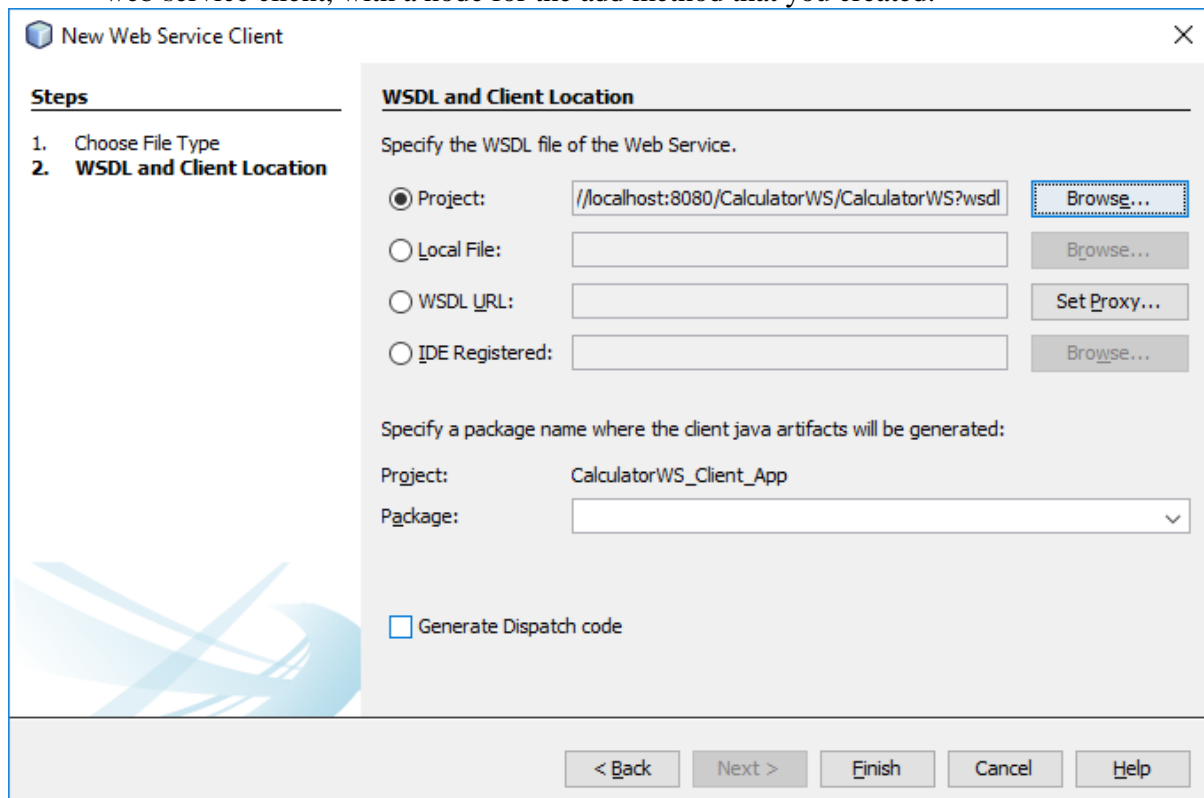
1. Choose File > New Project. Select Java Application from the Java category. Name the project CalculatorWS\_Client\_Application. Leave Create Main Class selected and accept all other default settings. Click Finish.



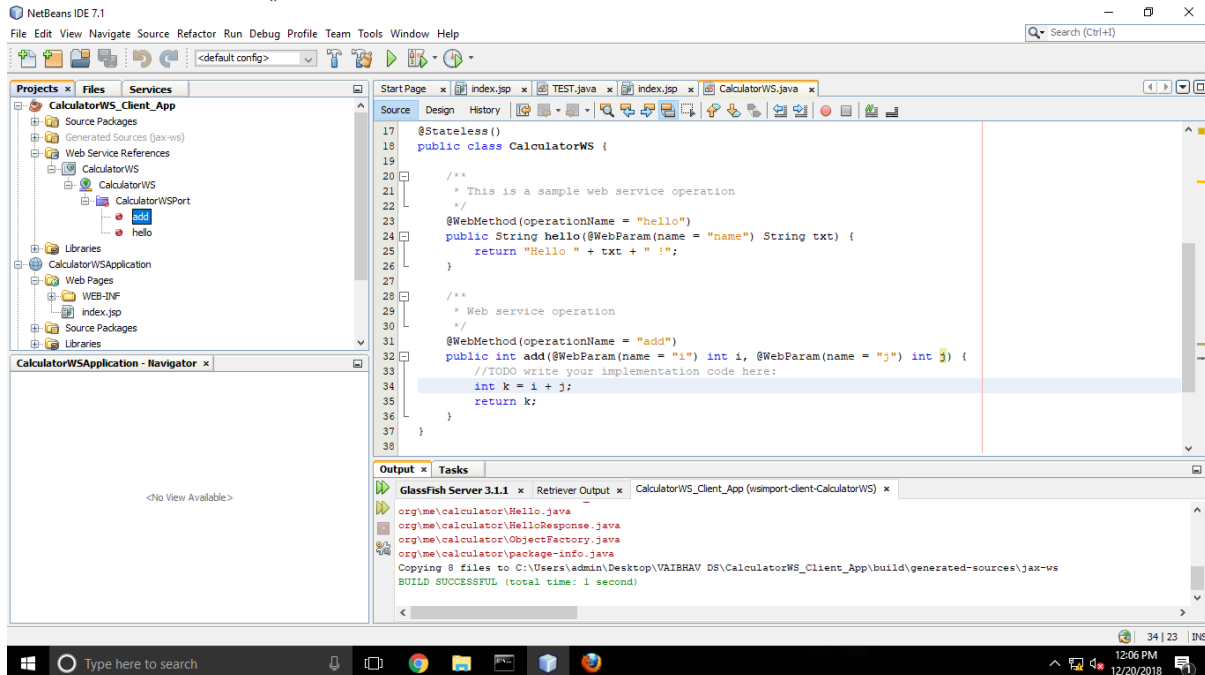
2. Right-click the CalculatorWS\_Client\_Application node and choose New > Web Service Client. The New Web Service Client wizard opens.



3. Select Project as the WSDL source. Click Browse. Browse to the CalculatorWS web service in the CalculatorWSApplication project. When you have selected the web service, click OK.
4. Do not select a package name. Leave this field empty.
5. Leave the other settings at default and click Finish. The Projects window displays the new web service client, with a node for the add method that you created:



6. Double-click your main class so that it opens in the Source Editor. Drag the add node below the main() method.



You now see the following:

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

```
{
```

```
// TODO code application logic here
```

```
}
```

```
private static int add(int i, int j)
```

```
{
```

```
org.me.calculator.CalculatorWS_Service service = new
```

```
org.me.calculator.CalculatorWS_Service();
```

```
org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
```

```
return port.add(i, j);
```

```
}
```

7. In the main() method body, replace the TODO comment with code that initializes values for i and j, calls add(), and prints the result.

Distributed Systems – Practical Manual – 2013-2014

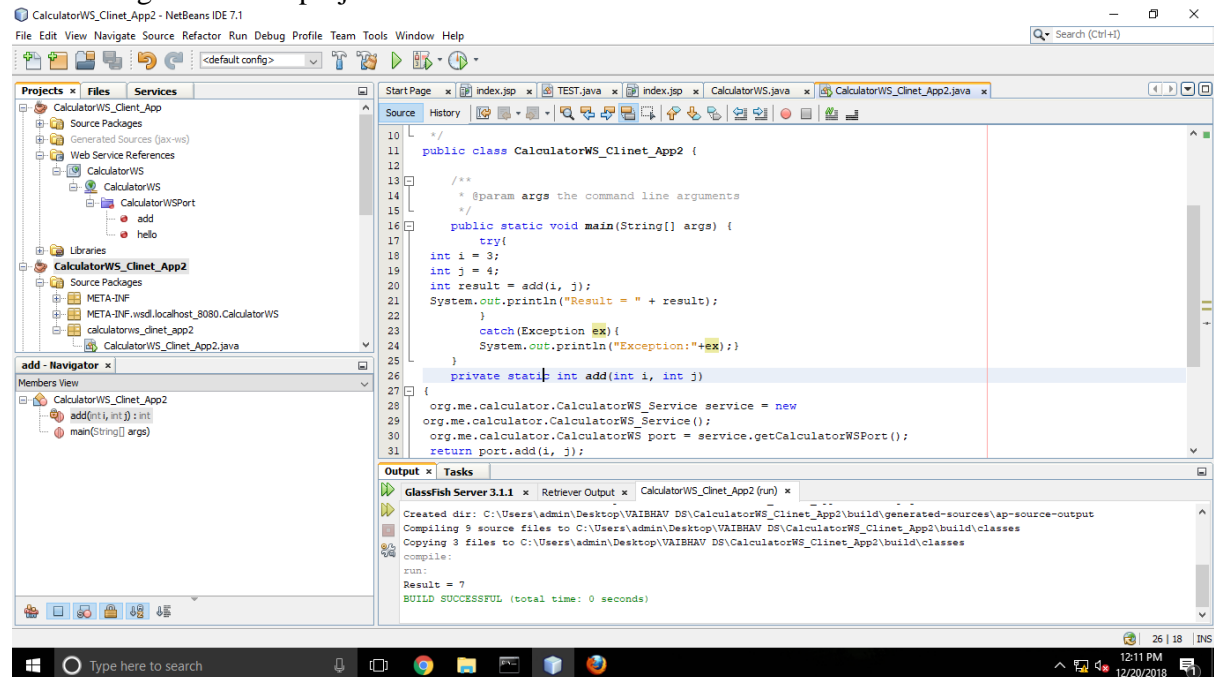
M.Sc. – I.T. – Semester I Page 37

```
public static void main(String[] args)
{ int i = 3; int j = 4; int result = add(i, j); System.out.println("Result = " + result);
}
```

8. Surround the main() method code with a try/catch block that prints an exception.

```
public static void main(String[] args)
{ try
{ int i = 3; int j = 4; int result = add(i, j); System.out.println("Result = " + result); } catch (Exception ex) { System.out.println("Exception: " + ex); }
}
```

10. Right-click the project node and choose Run.



The Output window now shows the sum:  
 compile:  
 run:  
 Result = 7  
 BUILD SUCCESSFUL (total time: 1 second)

### **Practical 5B:** Implementing Web Service that connects to MySQL database.

#### **1) Creating MySQL DB Table**

create database bookshop;

use bookshop;

✓ **Create a table named Books that will store valid books information**

create table books(isbn varchar(20) primary key, bookname varchar(100), bookprice varchar(10));

✓ **Insert valid records in the Books table**

insert into books values("111-222-333", "Learn My SQL", "250");



```
insert into books values("111-222-444","Java EE 6 for Beginners","850");  
insert into books values("111-222-555","Programming with Android","500");  
insert into books values("111-222-666","Oracle Database for you","400");  
insert into books values("111-222-777","Asp.Net for advanced programmers","1250");
```

## **2) Creating a web service**

### **i. Choosing a container**

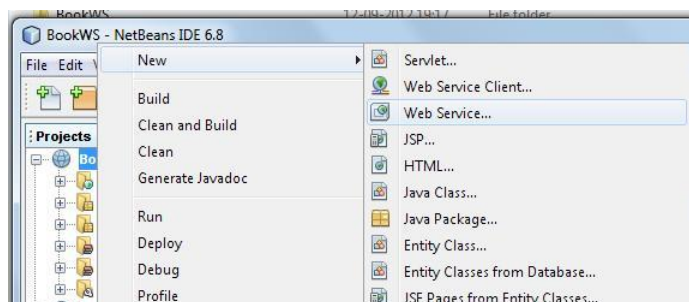
- ☐ Web service can be either deployed in a Web container or in an EJB container.
- ☐ If a Java EE 6 application is created, use a Web container because EJBs can be placed directly in a Web application.

### **ii. Creating a web application**

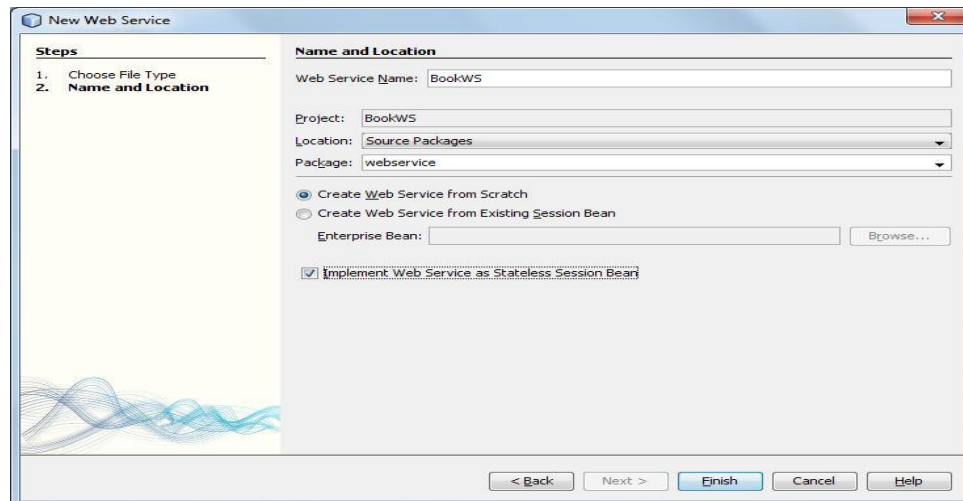
- ☐ To create a Web application, select File - New Project.
- ☐ New Project dialog box appears. Select Java Web available under the Categories section and Web Application available under the Projects section. Click Next.
- ☐ New Web Application dialog box appears. Enter BookWS as the project name in the Project Name textbox and select the option Use Dedicated Folder for Storing Libraries.
- Click Next. Server and Settings section of the New Web Application dialog box appears. Choose the default i.e. GlassFish v3 Domain as the Web server, the Java EE 6 Web as the Java EE version and the Context Path.
- ☐ Click -Finish
- ☐ The Web application named BookWS is created.

### **iii. Creating a web service**

- ☐ Right-click the BookWS project and select New -> Web Service as shown in diagram.



- ☐ New Web Service dialog box appears. Enter the name BookWS in the Web Service Name textbox, webservice in the Package textbox, select the option Create Web Service from scratch and also select the option implement web service as a stateless session bean as shown in the diagram.



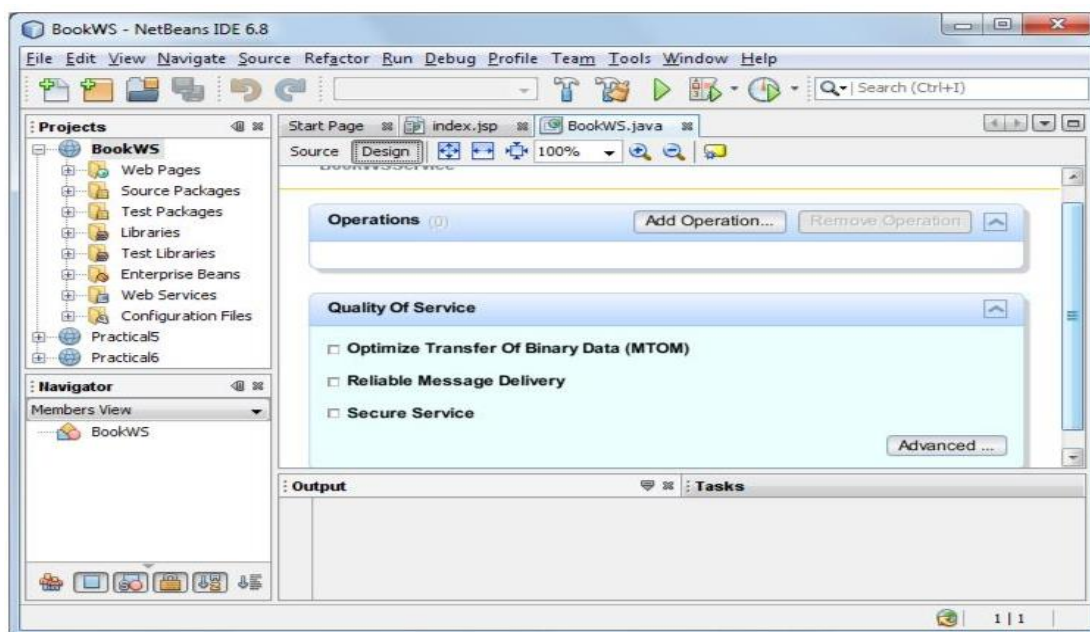
- ☐ Click Finish.
- ☐ The web service in the form of java class is ready.

### 3) Designing the web service

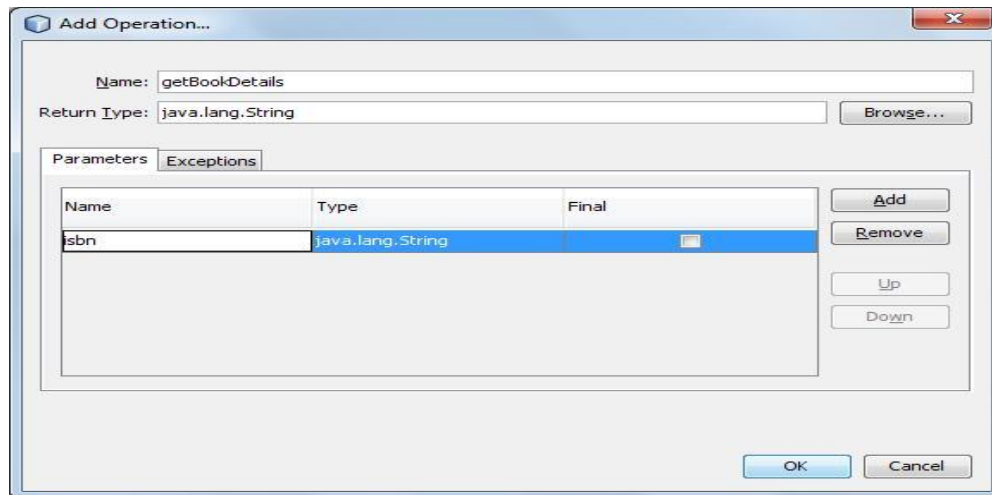
Now add an operation which will accept the ISBN number from the client to the web service.

#### i. Adding an operation to the web service

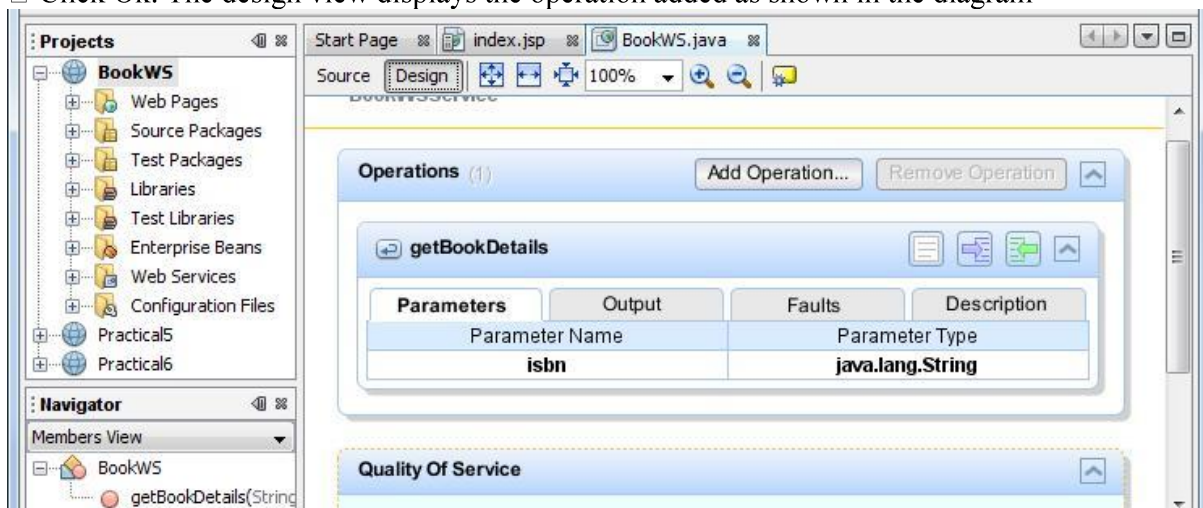
- ☐ Change the source view of the BookWS.java to design view by clicking Design available just below the name of the BookWS.java tab.
- ☐ The window changes as shown in the diagram.



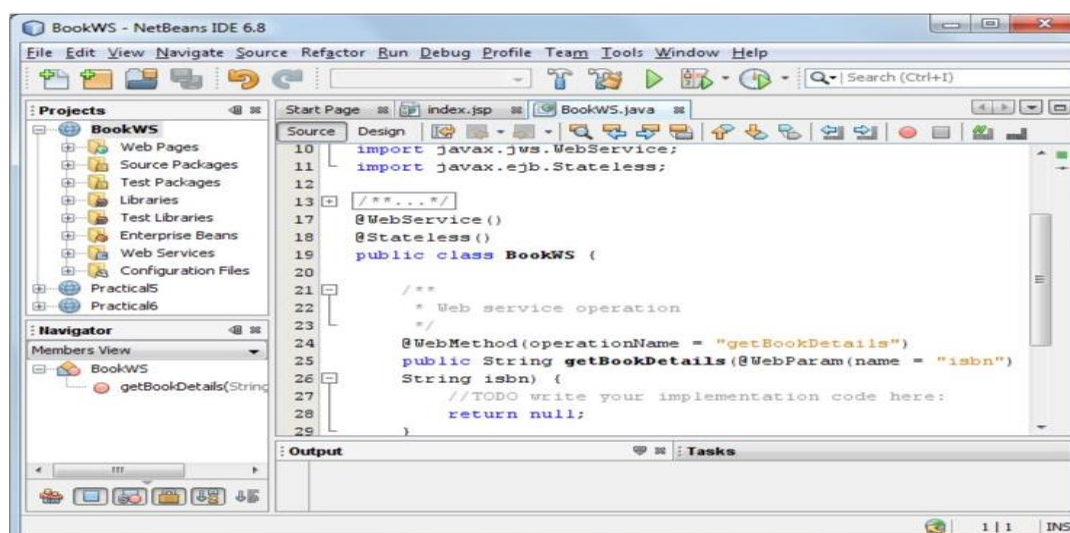
- Click Add Operation available in the design view of the web service.
  - ☐ Add Operation dialog appears. Enter the name getBookDetails in the Name textbox and java.lang.String in the Return Type textbox as shown in the diagram.
  - ☐ In Add Operation dialog box, click Add and create a parameter of the type String named isbn as shown in the diagram.



- Click Ok. The design view displays the operation added as shown in the diagram



- Click Source. The code spec expands due to the operation added to the web service as shown in the diagram.



- Modify the code spec of the web service BookWS.java.

**Code Spec**

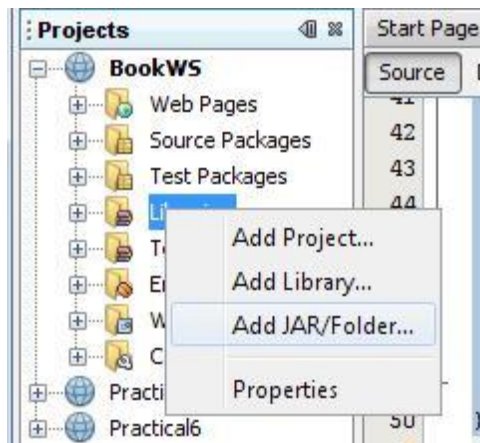
```
package webservice;
import java.sql.*;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.ejb.Stateless;
@WebService()
@Stateless()
public class BookWS {
/**
 * Web service operation
 */
@WebMethod(operationName = "getBookDetails")
public String getBookDetails(@WebParam(name = "isbn")
String isbn) {
//TODO write your implementation code here:
Connection dbcon = null;
Statement stmt = null;
ResultSet rs = null;
String query = null;
try
{
Class.forName("com.mysql.jdbc.Driver").newInstance();
dbcon = DriverManager.getConnection("jdbc:mysql://localhost/bookshop","root","123");
stmt = dbcon.createStatement();
query = "select * from books where isbn = " + isbn + "";
rs = stmt.executeQuery(query);
rs.next();
String bookDetails = "<h1>The name of the book is <b>" + rs.getString("bookname") + "</b>"
and its cost is <b>" + rs.getString("bookprice") + "</b></h1>.";
return bookDetails;
}
catch(Exception e)
{
System.out.println("Sorry failed to connect to the database.." + e.getMessage());
}
return null;
}
}
```

**Explanation**

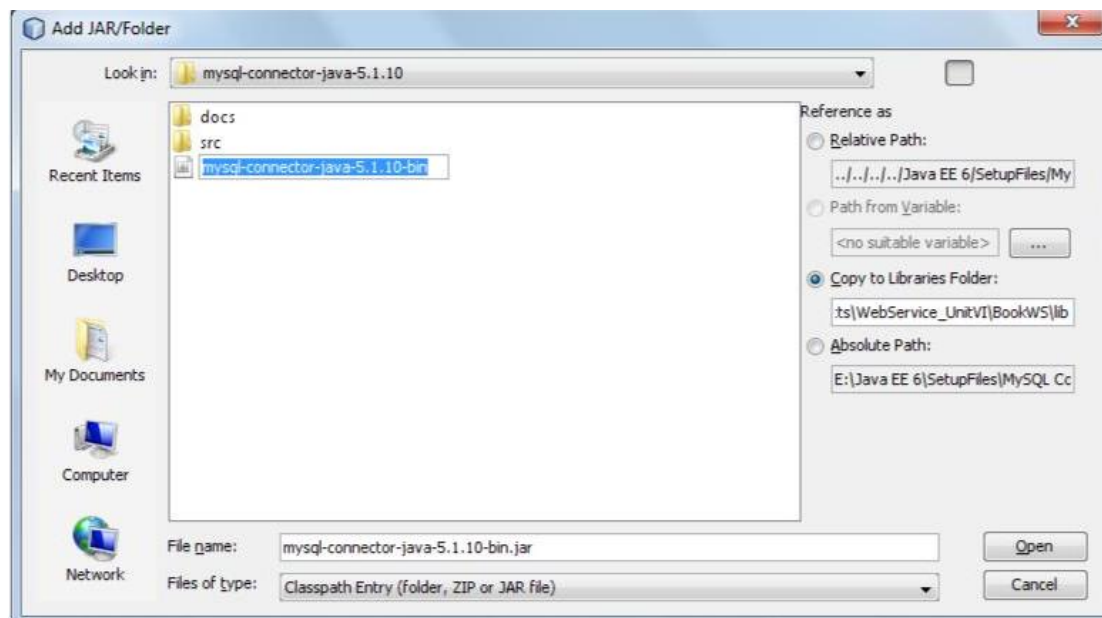
☐ In the above code spec, a database connection is established. Based on the ISBN number entered by the user, the associated book name and price is retrieved and returned.

**4) Adding the MySQL connector**

- ☐ We need to add a reference of MySQL connector to our web service. It is via this connector that our web service will be able to communicate with the database.
- ☐ Right click on the libraries and select Add JAR/Folder as shown in the diagram.



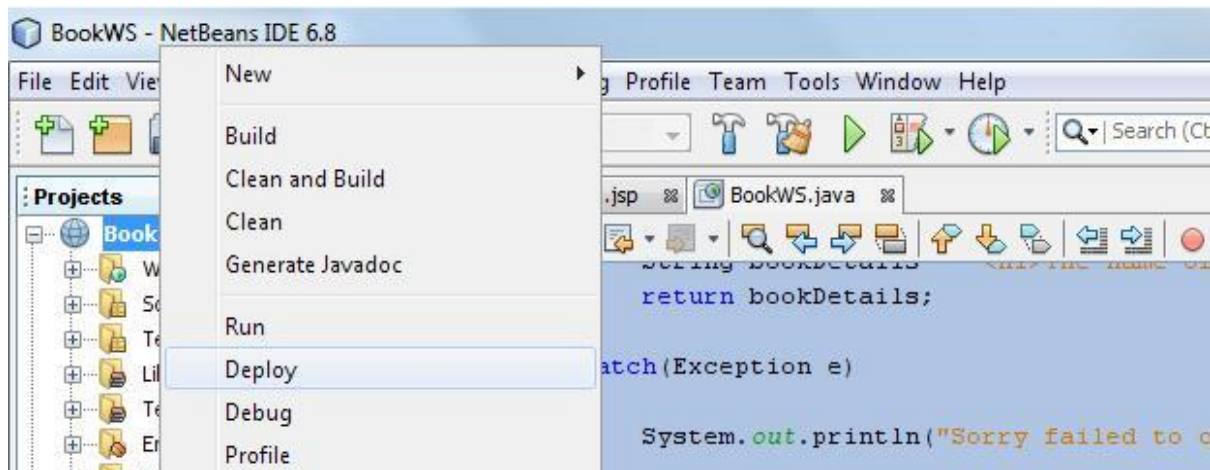
□ Choose the location where mysql-connector-java-5.1.10-bin is located, select it and click on open as shown.



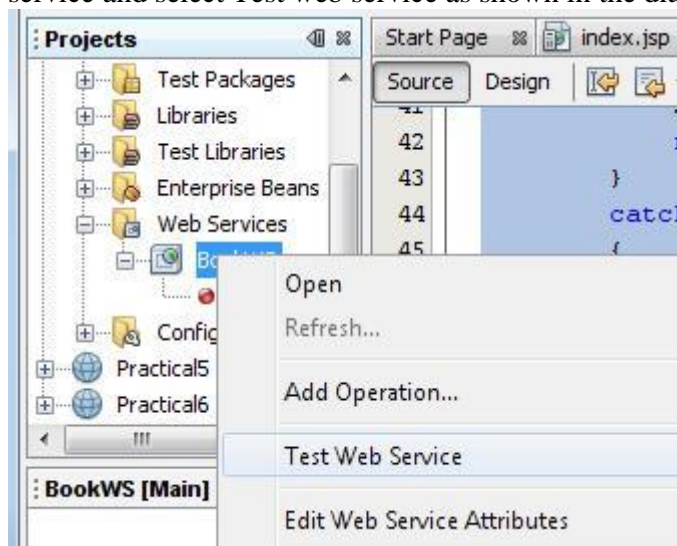
### 5) Deploying and testing the web service

- When a web service is deployed to a web container, the IDE allows testing the web service to see if it functions as expected.
- The tester application provided by GlassFish, is integrated into the IDE for this purpose as it allows the developer to enter values and test them.
- No facility for testing whether an EJB module is deployed successfully is currently available.
- To test the BookWS application, right click the BookWS project and select Deploy as shown in the diagram.





- ☐ The IDE starts the server, builds the application and deploys the application to the server.
- ☐ Follow the progress of these operations in the BookWS (run-deploy) and GlassFish v3 Domain tabs in the Output view.
- ☐ Now expand the web services directory of the BookWS project, right-click the BookWS Web service and select Test web service as shown in the diagram.



- ☐ The IDE opens the tester page in the web browser, if the web application is deployed using GlassFish server as shown in the figure.



- Enter the ISBN number as shown in the diagram.
- Click getBookDetails. The book name and its cost are displayed as shown in the diagram.

← → ↻
localhost:35637/BookWS/BookWSService?Tester

## getBookDetails Method invocation

---

**Method parameter(s)**

| Type             | Value       |
|------------------|-------------|
| java.lang.String | 111-222-333 |

---

**Method returned**

java.lang.String : "The name of the book is Learn My SQL and its cost is 250<."

---

**SOAP Request**

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:getBookDetails xmlns:ns2="http://webservice/">
      <isbn>111-222-333</isbn>
    </ns2:getBookDetails>
  </S:Body>
</S:Envelope>
```

---

**SOAP Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

**6) Consuming the web service**

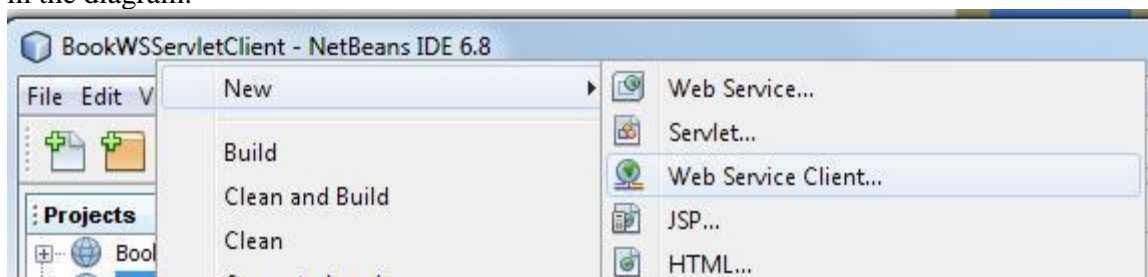
☐ Once the web service is deployed, the next most logical step is to create a client to make use of the web service's `getBookDetails()` method.

**i. Creating a web application**

- ☐ To create a web application, select File -> New Project.
- ☐ New project dialog box appears, select java web available under the categories section and web application available under the projects section. Click Finish.
- ☐ New web application dialog box appears. Enter BookWSServletClient as the project name in the Project Name textbox and select the option Use Dedicated Folder for Storing Libraries.
- ☐ Click Next. Server and settings section of the new web application, dialog box appears. Choose the default i.e. GlassFish v3 Domain as the web server, the Java EE 6 web as the Java EE version and the context path.
- ☐ Click Finish.
- ☐ The web application named BookWSServletClient is created.

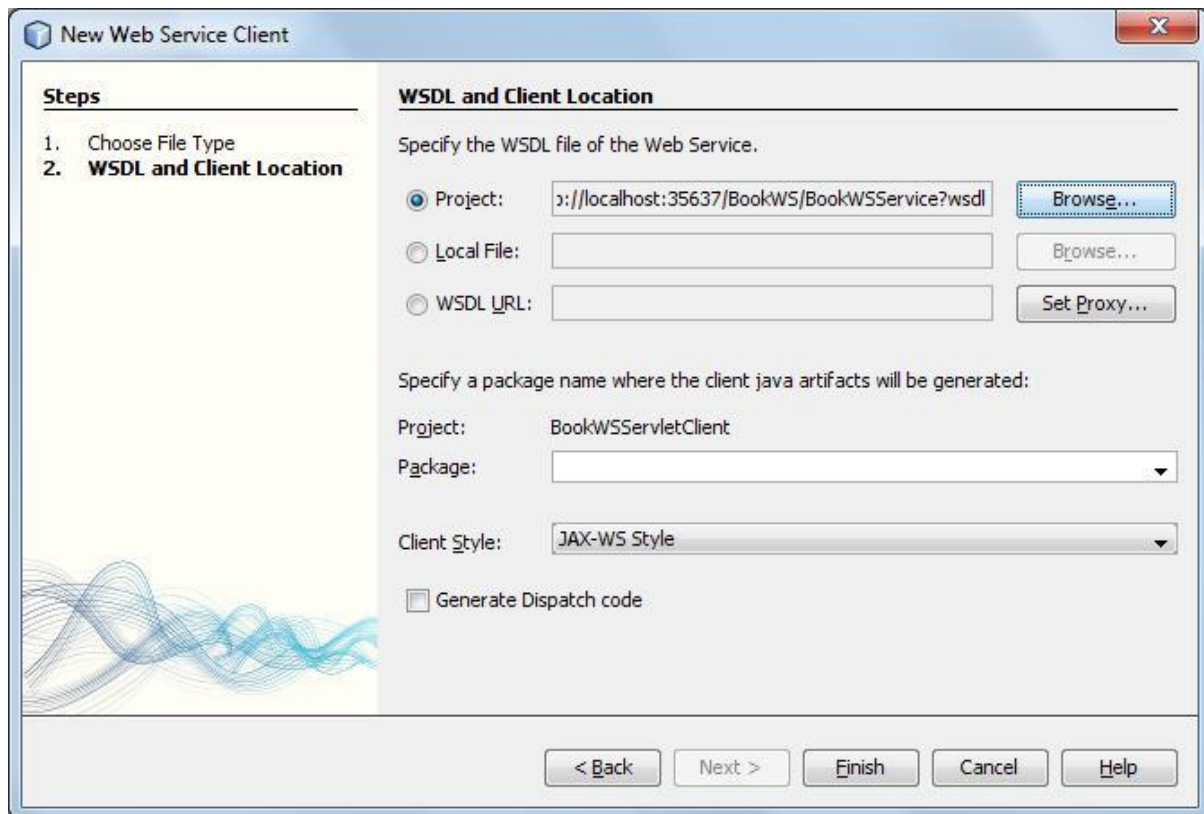
**ii. Adding the web service to the client application**

☐ Right-click the BookWSServletClient project and select New -> Web Service Client as shown in the diagram.



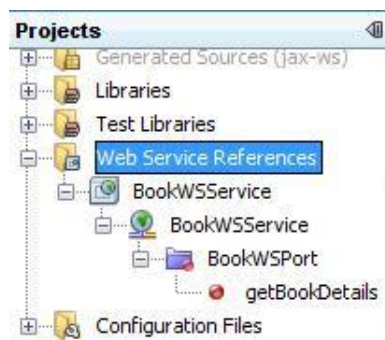
☐ New Web Service Client dialog box appears. In the Project section, click Browse and browse through the web service which needs to be consumed. Click ok. The name of the web service appears in the New Web Service Client as shown in the diagram.





- ☐ Leave the other settings as it is. Click Finish.

The Web Service Reference directory is added to the BookWSServletClient application as shown in the diagram. It displays the structure of the newly created client including the getBookDetails() method created earlier.

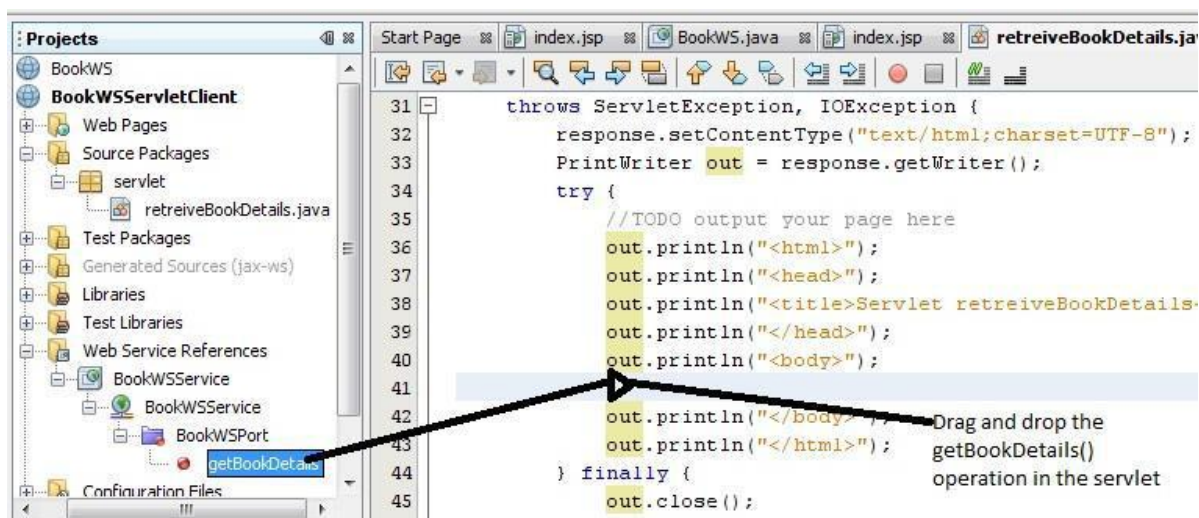


### iii. Creating a servlet

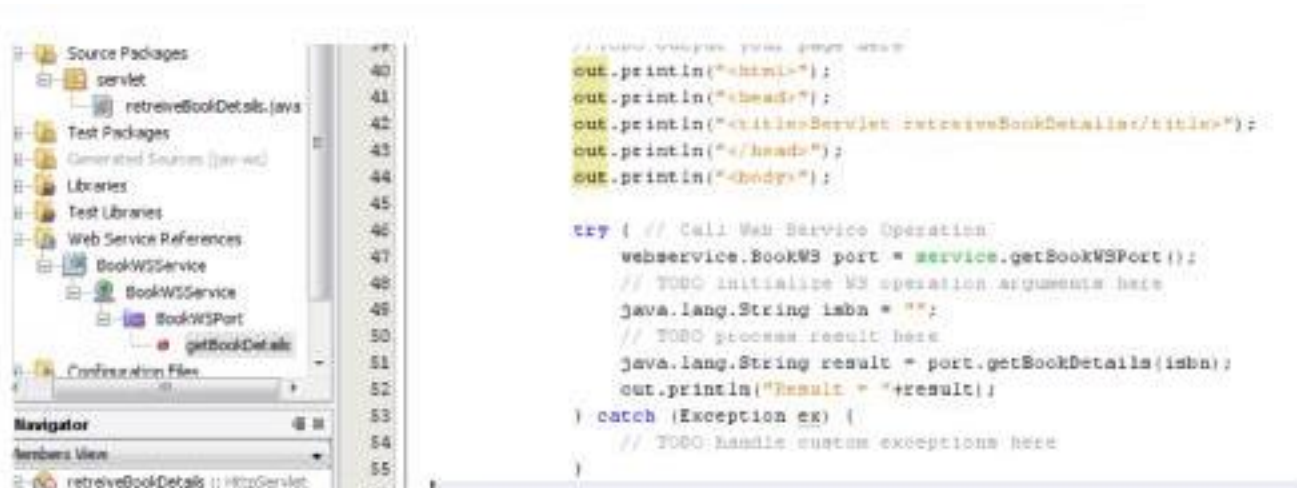
- ☐ Create retrieveBookDetails.java using NetBeans IDE.
- ☐ Right click source package directory, select New -> Servlet.
- ☐ New Servlet dialog box appears. Enter retrieveBookDetails in the Class Name textbox and enter servlet in the package textbox.

- ☐ Click Next. Configure Servlet Deployment section of the New Servlet dialog box appears. Keep the defaults.
- ☐ Click Finish.
- ☐ This creates the servlet named retrieveBookDetails.java in the servlet package.
- ☐ retrieveBookDetails.java is available with the default skeleton created by the NetBeans IDE which needs to be modified to hold the application logic.
- ☐ In the retrieveBookDetails.java source code, remove the following comments available in the body of the processRequest() method.  
/\*TODO output your page here\*/
- ☐ Replace the following code spec:

out.println("<h1>Servlet retrieveBookDetails at " + request.getContextPath () + "</h1>");  
With the code spec of the getBookDetails() operation of the web service by dragging and dropping the getBookDetails operation as shown in the diagram.



- ☐ The Servlet code spec changes as shown in the diagram.



- ☐ The web service is instantiated by the @WebServiceRef annotation.
- ☐ Now change the following code spec:

```
java.lang.String isbn = "";  
to  
java.lang.String isbn = request.getParameter("isbn");
```

#### **iv. Creating an HTML form**

- ☐ Once the web service is added and the servlet is created, the form to accept ISBN from the user needs to be coded.
- ☐ Since NetBeans IDE by default [as a part of Web Application creation] makes available index.jsp file. Modify it to hold the following code spec.

```
<% @page contentType="text/html" pageEncoding="UTF-8"% >  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>SOAP Cleint - Get Book Details</title>  
</head>  
<body bgcolor="pink">  
<form name="frmgetBookDetails" method="post" action="retreiveBookDetails">  
<h1>  
ISBN : <input type="text" name="isbn"/><br><br>  
</h1>  
<input type="submit" value="Submit"/>  
</form>  
</body>  
</html>
```

#### **v. Building the Web Application**

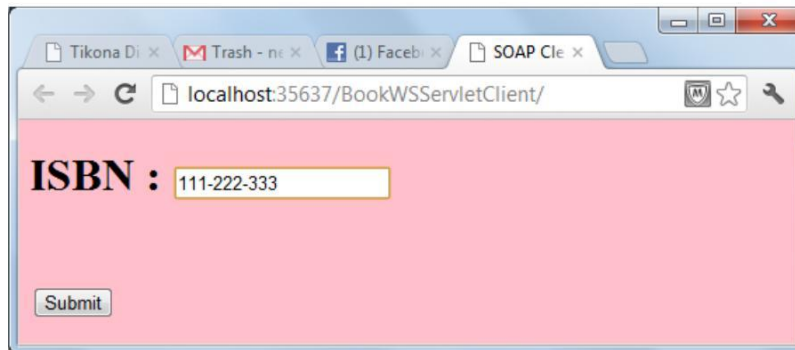
- ☐ Build the web application.
- ☐ Right click BookWSServletClient project and select Build.
- ☐ Once the Build menu item is clicked the details about the compilation and building of the BookWSServletClient Web application appears in the output – BookWSServletClient (dist) window.

#### **vi. Running the Application**

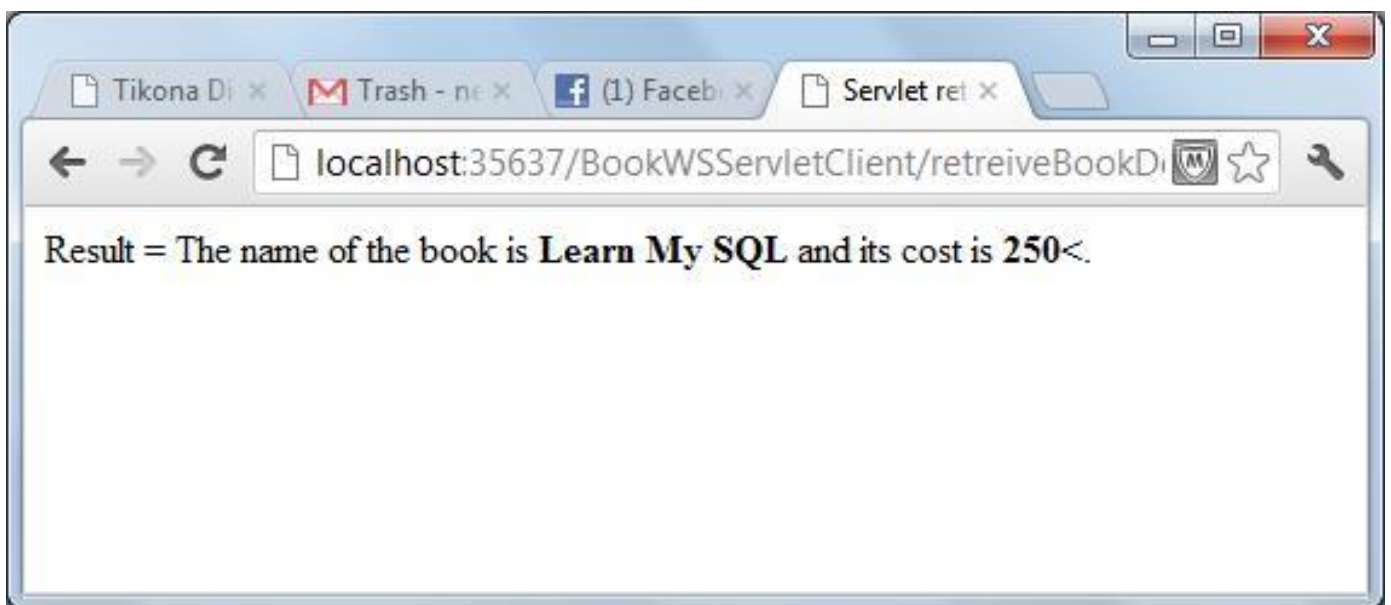
- ☐ Once the compilation and building of the web application is done run the application.
- ☐ Right click the BookWSServerCleint project and select run.
- ☐ Once the run processing completes in NetBeans IDE a web browser is automatically launched and the BookWSServletCleint application is executed as shown in the diagram.



- Enter the ISBN as shown in the diagram.



- Click Submit. The book name and its cost are displayed as shown in the diagram.



This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

**Practical No: 06**

**Aim:** Write a program to execute any one mutual exclusion algorithm.(Token Ring)

**Code:**

**TokenClient1.java**

```
import java.io.*;
import java.net.*;
public class TokenClient1 {
    public static void main(String arg[]) 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(9001);
            tkcl.setSendPort(9004);
            tkcl.setRecPort(8002);
            lclhost=InetAddress.getLocalHost();
            tkser.setSendPort(9000);
            if(tkcl.hasToken == true)
            {
                System.out.println("Do you want to enter the Data --> YES/NO");
                br=new BufferedReader(new InputStreamReader(System.in));
                str=br.readLine();
                if(str.equalsIgnoreCase("yes"))
                {
                    System.out.println("ready to send");
                    tkser.setSendData = true;
                    tkser.sendData();
                    tkser.setSendData = false;
                }
                else if(str.equalsIgnoreCase("no"))
                {
                    System.out.println("i m in else");
                    //tkcl.hasToken=false;
                    tkcl.sendData();
                    tkcl.recData();
                    System.out.println("i m leaving else");
                }
            }
        }
    }
}
```

```

    }
    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("sending ");
            System.out.println("Enter the Data");
            br=new BufferedReader(new InputStreamReader(System.in));
            str = "ClientOne....." + br.readLine();
            System.out.println("now sending");
        }
        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 buffer[] = new byte[256];
        DatagramSocket ds;
        DatagramPacket dp;
        ds = new DatagramSocket(recport);
        dp = new DatagramPacket(buffer,buffer.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 arg[]) 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);
            System.out.println("entering if");
            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("ignorecase");
                    ser.setSendData = true;
                    ser.sendData();
                }
                else if(str1.equalsIgnoreCase("no"))    {
                    tkcl.sendData();
                    hasToken=false;
                }
            }
            else    {
                System.out.println("entering recieving 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    {
            System.out.println("case");
            BufferedReader br;
            String str="Token";
            DatagramSocket ds;
            DatagramPacket dp;
            if(setSendData == true)    {
                System.out.println("Enter the Data");
                br=new BufferedReader(new InputStreamReader(System.in));
                str = "ClientTwo....." + br.readLine();    }
            ds = new DatagramSocket(sendport);
            dp = new DatagramPacket(str.getBytes(),str.length(),lclhost,sendport-1000);
            ds.send(dp);
            ds.close();
            System.out.println("Data Sent");
            setSendData = false;
            hasToken = false;
        }
        void recData()throws Exception    {
            String msgstr;
            byte buffer[] = new byte[256];
            DatagramSocket ds;
            DatagramPacket dp;
            ds = new DatagramSocket(recport);
            //ds = new DatagramSocket(4000);
            dp = new DatagramPacket(buffer,buffer.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;
    }
}
}
```

**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 buff[]=new byte[256];
            DatagramSocket ds;
            DatagramPacket dp;
            String str;
            ds=new DatagramSocket(recport);
            dp=new DatagramPacket(buff,buff.length);
            ds.receive(dp);
            ds.close();
            str=new String(dp.getData(),0,dp.getLength());
            System.out.println("The message is "+str);
        }
    }
}
```

**Output:**

```

C:\Users\Aditya>cd desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd "Token Ring"
C:\Users\Aditya\Desktop\DS\Token Ring>javac TokenServer.java
C:\Users\Aditya\Desktop\DS\Token Ring>java TokenServer
The message is ClientOne.....hello
The message is ClientTwo.....hi everyone

```

```

C:\Users\Aditya>cd desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd "Token Ring"
C:\Users\Aditya\Desktop\DS\Token Ring>javac TokenClient1.java
C:\Users\Aditya\Desktop\DS\Token Ring>java TokenClient1
Do you want to enter the Data --> YES/NO
yes
ready to send
sending
Enter the Data
hello
now sending
Do you want to enter the Data --> YES/NO
no
i m in else

```

```
Command Prompt - java TokenClient2

-version          Version information
-help            Print a synopsis of standard options
-Akey[=value]     Options to pass to annotation processors
-X              Print a synopsis of nonstandard options
-J<flag>         Pass <flag> directly to the runtime system

C:\Users\Aditya>cd desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS>cd "Token Ring"
C:\Users\Aditya\Desktop\DS\Token Ring>javac TokenClient2.java
C:\Users\Aditya\Desktop\DS\Token Ring>java TokenClient2
entering if
entering recieving mode
The data is Token
entering if
Do you want to enter the Data --> YES/NO
yes
ignorecase
case
Enter the Data
hii everyone
Data Sent
entering if
Do you want to enter the Data --> YES/NO
```

[illegible]

**Practical No: 07**

**Aim:** Write a program to implement any one election algorithm.

**Code:**

**1. Election.c**

```
#include<stdio.h>
#include<conio.h>
#include<process.h>

struct proc
{
int live;
int identifier;
}
process[10];
int n,cordinator=1;
/***** DISPLAY PROCESSES *****/
void display()
{
int i;
printf("\n PROCESSES ARE\n\n");
printf("Processes ");
for(i=1;i<=n;i++)
{
printf("P%d\t",i);
}
printf("\nlive ");
for(i=1;i<=n;i++)
{
printf("%d\t",process[i].live);
}
printf("\nidentifier ");
for(i=1;i<=n;i++)
{
printf("%d\t",process[i].identifier);
}
}
/***** BULLY ALGORITHM *****/
void bully()
{
int ch,c,id,i=0,cordinator,init,max=-99;
cordinator=i;
for(i=1;i<=n;i++)
{
if(process[cordinator].identifier<process[i].identifier&& process[i].live==1)
cordinator=i;
}
}
```

```
printf("\n\n CURRENT CO-ORDINATOR IS=P%d",cordinator);
while(ch!=4)
{
printf("\n\n\n *** BULLY ALGORITHM ***");
printf("\n1.Crash a Process\n2.Activate Process\n3.Display\n4.Exit");
printf("\nENTER UR CHOICE");
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("\n Enter the process id to crash");
scanf("%d",&id);
if(process[id].live==0)
{
printf("\n Already crashed process");
}
else
{
process[id].live=0;
printf("\n process P%d is crashed",id);
if(id==cordinator)
{
while(1)
{
printf("\n Enter process id who intiates election");
scanf("%d",&init);
if(process[init].live==0)
{
printf("\n the selected process is crashed");
}
else
{
for(i=1;i<=n;i++)
{
if(i!=init&& process[i].identifier>process[init].identifier)
printf("\n Election MSG sent from %d to %d",init,i);
}
for(i=1;i<=n;i++)
{
if(i!=init)
{
if(process[i].identifier>process[init].identifier&&process[i].live!=0)
{
printf("\n OK from %d to %d",i,init);
}
}
}
for(i=1;i<=n;i++)
{
if(max<process[i].identifier && process[i].live!=0)
```

```
{
cordinator=i;

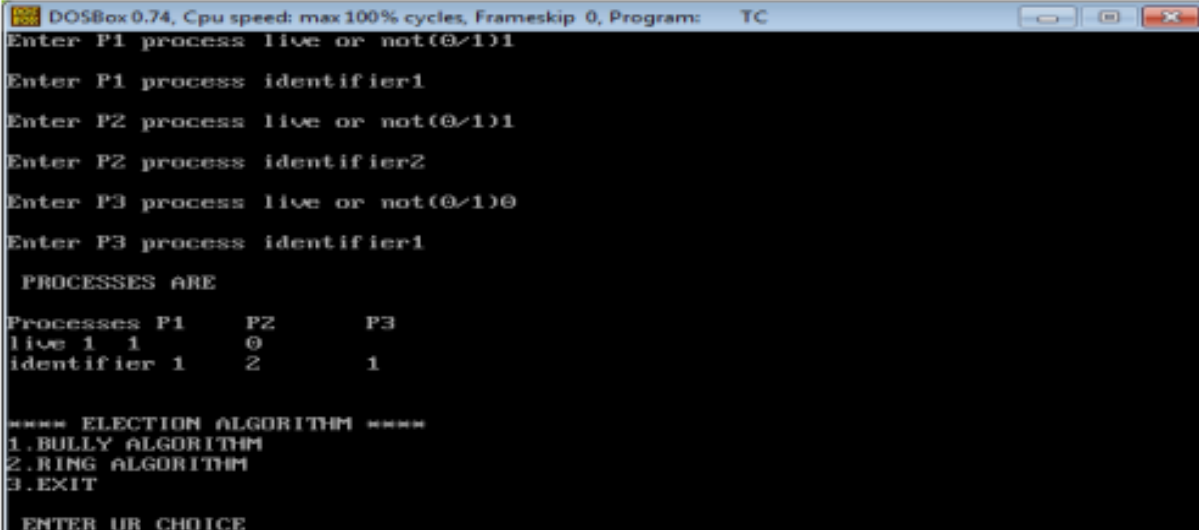
max=process[i].identifier;
}
}
printf("\n\n NEW CO-ORDINATOR IS=P%d",cordinator);
break;
}
}
}
}
break;
case 2:
printf("\n Enter process id to activate");
scanf("%d",&id);
if(process[id].live==1)
{
printf("\n Process %d is already active",id);
}
else
{
process[id].live=1;
printf("\n Process %d activated",id);
}
if(process[id].identifier>process[cordinator].identifier)
{
cordinator=id;
printf("\n NEW CO-ORDINATOR IS=P%d\n\n",id);
}
break;
case 3:
display();
break;
case 4:
break;
}
}
}
/***** RING ALGORITHM *****/
void ring()
{
int ch,c,id,i=0,init,max=-99,last;
for(i=1;i<=n;i++)
{
if(process[cordinator].identifier<process[i].identifier&&process[i].live==1)
cordinator=i;
}
printf("\n\n CURRENT CO-ORDINATOR IS=P%d",cordinator);
while(ch!=4)
```



```
{
printf("\n\n *** RING ALGORITHM ***");
printf("\n1.Crash a Process\n2.Activate Process\n3.Display\n4.Exit");
printf("\nENTER UR CHOICE");
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("\n Enter the process id to crash");
scanf("%d",&id);
if(process[id].live==0)
{
printf("\n Already crashed process");
}
else
{
process[id].live=0;
printf("\n process P%d is crashed",id);
if(id==cordinator)
{
while(1)
{
printf("\n Enter process id who intiates election");
scanf("%d",&init);
if(process[init].live==0)
{
printf("\n the selected process is crashed");
}
else
{
last=init;
printf("\nElection MSG sent from =%d",last);
for(i=init+1;i<=n;i++)
{
if(i!=init)
printf(" ->%d",i);
}
for(i=1;i<init;i++)
{
if(i!=init)
printf(" ->%d",i);
last=i;
}
for(i=init+1;i<=n;i++)
{
if(max<process[i].identifier && process[i].live==1)
{
cordinator=i;
max=process[i].identifier;
}
}
}
}
```

```
}
for(i=1;i<=init;i++)
{
if(max<process[i].identifier && process[i].live==1)
{
cordinator=i;
max=process[i].identifier;
}
}
printf("\n\n NEW COORDINATOR
IS=P%d",cordinator);
break;
}
}
}
}
break;
case 2:
printf("\n Enter process id to activate");
scanf("%d",&id);
if(process[id].live==1)
{
printf("\n Process %d is already active",id);
}
else
{
process[id].live=1;
printf("\n Process %d activated",id);
if(process[id].identifier>process[cordinator].identifier)
{
printf("\n NEW CO-ORDINATOR
IS=P%d\n\n",id);
cordinator=id;
}
}
break;
case 3:
display();
break;
case 4:
break;
}
}
}
void main()
{
int ch,i,c;
clrscr();
printf("\n ENTER NO. OF PROCESSES");
scanf("%d",&n);
```

```
for(i=1;i<=n;i++)
{
printf("\nEnter P%d process live or not(0/1)",i);
scanf("%d",&process[i].live);
printf("\nEnter P%d process identifier",i);
scanf("%d",&process[i].identifier);
}
display();
while(1)
{
printf("\n\n**** ELECTION ALGORITHM ****");
printf("\n1.BULLY ALGORITHM\n2.RING ALGORITHM\n3.EXIT");
printf("\n\n ENTER UR CHOICE");
scanf("%d",&ch);
switch(ch)
{
case 1:
bully();
break;
case 2:
ring();
break;
case 3:
exit(0);
}
}
}
```

**Output:**

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter P1 process live or not(0/1)1
Enter P1 process identifier1
Enter P2 process live or not(0/1)1
Enter P2 process identifier2
Enter P3 process live or not(0/1)0
Enter P3 process identifier1
PROCESSES ARE
Processes P1    P2    P3
live 1 1      0
identifier 1    2    1

**** ELECTION ALGORITHM ****
1.BULLY ALGORITHM
2.RING ALGORITHM
3.EXIT
ENTER UR CHOICE
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

**Practical No: 08**

**Aim:** Show the implementation of any one clock synchronization algorithm.

**Code:**

**SCClient.java**

```
import java.io.*;
import java.net.*;
public class SCClient
{
    public static void main(String args[])throws Exception
    {
        InetAddress lclhost;
        lclhost=InetAddress.getLocalHost();
        while(true)
        {
            Client cntl=new Client(lclhost);
            cntl.sendPort(9001);
            cntl.sendData();
        }}
    class Client
    {
        InetAddress lclhost;
        int senport;
        Client(InetAddress lclhost)
        {
            this.lclhost=lclhost;
        }
        void sendPort(int senport)
        {
            this.senport=senport;
        }
        void sendData()throws Exception
        {
            DatagramPacket dp;
            DatagramSocket ds;
            BufferedReader br;
            br=new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter the data");
            String str=br.readLine();
            ds = new DatagramSocket(senport);
            dp = new DatagramPacket(str.getBytes(),str.length(),lclhost,senport-1000);
            ds.send(dp);
            ds.close();
        }}
}
```

**SCServer.java**

```
import java.io.*;
import java.net.*;
import java.sql.*;
public class SCServer
{
    public static void main(String args[])throws Exception
    {
        InetAddress lclhost;
        lclhost=InetAddress.getLocalHost();
        long maxtime,skewtime,datatime;
        String maxtimestr,skewtimestr;
        BufferedReader br;
        CIntServer ser=new CIntServer(lclhost);
        System.out.println("Enter the maximum time");
        br = new BufferedReader(new InputStreamReader(System.in));
        maxtimestr=br.readLine();
        System.out.println("Enter the maximum skew time");
        br = new BufferedReader(new InputStreamReader(System.in));
        skewtimestr=br.readLine();
        maxtime=Long.parseLong(maxtimestr);
        skewtime=Long.parseLong(skewtimestr);
        while(true)
        {
            datatime = System.currentTimeMillis();
            long G = datatime-maxtime-skewtime;
            System.out.println("G =" +G);
            ser.setTimeStamp(new Timestamp(G));
            ser.recPort(8001);
            ser.recData();
        }
    }
    class CIntServer
    {
        InetAddress lclhost;
        int recport;
        Timestamp obtmp;
        CIntServer(InetAddress lclhost)
        {
            this.lclhost = lclhost;
        }
        void recPort(int recport)
        {
            this.recport = recport;
        }
        void setTimeStamp(Timestamp obtmp)
        {

```

```
this.obtmp = obtmp;
}
void recData()throws Exception
{
String msgstr="";
DatagramSocket ds;
DatagramPacket dp;
BufferedReader br;
byte buf[] = new byte[256];
ds = new DatagramSocket(recport);
dp = new DatagramPacket(buf,buf.length);
ds.receive(dp);
ds.close();
msgstr = new String(dp.getData(),0,dp.getLength());
System.out.println(msgstr);
Timestamp obtmp = new Timestamp(Long.parseLong(msgstr));
if(this.obtmp.before(obtmp) == true)
{
System.out.println("The Message is accepted");
}
else
{
System.out.println("The Message is rejected");
}}}
```

### Output:

```
Select Command Prompt - java SCServer

C:\Users\Aditya\Desktop\DS\clock>javac SCServer.java
C:\Users\Aditya\Desktop\DS\clock>java SCServer
Enter the maximum time
132
Enter the maximum skew time
45
G =1545221304050
123
The Message is rejected
G =1545221312878
5555555533333333333
The Message is accepted
G =1545221352426
```

```
Command Prompt - java SCClient

C:\Users\Aditya>cd Desktop
C:\Users\Aditya\Desktop>cd ds
C:\Users\Aditya\Desktop\DS> cd clock
C:\Users\Aditya\Desktop\DS\clock>javac SCClient.java
C:\Users\Aditya\Desktop\DS\clock>java SCClient
Enter the data
3356789
Enter the data
22
Enter the data
55555522222222222222
Enter the data
123
Enter the data
5555555533333333333
Enter the data
555333
Enter the data
123
Enter the data
```



[illegible]

## **Practical No: 9**

**Aim:** Implement the concept of distributed file system architecture.

### **Network File System:**

Goto root mode.

\$ su

password – fedora

Install nfs-utils

#yum install nfs-utils

Specify what we want to share

#/etc/exports

Start the Services

#systemctl start rpcbind.service

#systemctl start nfs-lock.service

#systemctl start nfs-service.service

#systemctl start nfs-idmap.service

#systemctl start mountd.service

Enable the Services

#systemctl enable rpcbind.service

#systemctl enable nfs-lock.service

#systemctl enable nfs-service.service

#systemctl enable nfs-idmap.service

#systemctl enable mountd.service

Type the message

cat>>vaibhav

Hello World!

Cat>>/nfstest/vaibhav

Hello Universe!

mkdir vsit

mount -t nfs 127.0.0.1:/nfstest vsit

**Output:**

```

user@localhost:/home/user
File Edit View Search Terminal Help
[user@localhost ~]$ yum install nfs-utils
Loaded plugins: langpacks, presto, refresh-packagekit
You need to be root to perform this command.
[user@localhost ~]$ su
Password:
su: incorrect password
[user@localhost ~]$ su
Password:
[root@localhost user]# yum install nfs-utils
Loaded plugins: langpacks, presto, refresh-packagekit
Existing lock /var/run/yum.pid: another copy is running as pid 1976.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: PackageKit
    Memory : 222 M RSS (628 MB VSZ)
    Started: Wed Dec 12 11:42:31 2018 - 00:42 ago
    State   : Sleeping, pid: 1976
Resolving Dependencies
--> Running transaction check
---> Package nfs-utils.x86_64 1:1.2.6-14.fc18 will be updated
---> Package nfs-utils.x86_64 1:1.2.7-6.fc18 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version           Repository        Size
=====
Updating:
nfs-utils              x86_64        1:1.2.7-6.fc18    updates           352 k
=====
Transaction Summary
=====
Upgrade 1 Package

Total download size: 352 k
Is this ok [y/N]: y
Downloading Packages:
Setting up and reading Presto delta metadata
warning: /var/cache/yum/x86_64/18/updates/packages/nfs-utils-1.2.7-6.fc18.x86_64
.rpm: Header V3 RSA/SHA256 Signature, key ID de7f38bd: NOKEY
Public key for nfs-utils-1.2.7-6.fc18.x86_64.rpm is not installed
nfs-utils-1.2.7-6.fc18.x86_64.rpm                                | 352 kB  00:00
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-x86_64
Importing GPG key 0xDE7F38BD:
  Userid      : "Fedora (18) <fedora@fedoraproject.org>"
  Fingerprint: 7efb 8811 dd11 e380 b679 fced ff01 125c de7f 38bd
  Package     : fedora-release-18-1.noarch (@anaconda)
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-x86_64

```

[illegible]

```
user@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost user]# systemctl enable nfs-lock.service
[root@localhost user]# systemctl enable nfs-server.service
ln -s '/usr/lib/systemd/system/nfs-server.service' '/etc/systemd/system/multi-user.target.wants/nfs-server.service'
[root@localhost user]# systemctl enable nfs-server.service
[root@localhost user]# systemctl enable nfs-idmap.service
ln -s '/usr/lib/systemd/system/nfs-idmap.service' '/etc/systemd/system/nfs.target.wants/nfs-idmap.service'
[root@localhost user]# systemctl enable nfs-idmap.service
[root@localhost user]# systemctl enable nfs-mountd.service
ln -s '/usr/lib/systemd/system/nfs-mountd.service' '/etc/systemd/system/nfs.target.wants/nfs-mountd.service'
[root@localhost user]# systemctl enable nfs-mountd.service
[root@localhost user]# cat >> vaibhav
Hello Snakes !^C
[root@localhost user]# cat >> /nfstest/vaibhav
Hello Snakes !
Hiss
^C
[root@localhost user]# mkdir vsit
[root@localhost user]# mount -t nfs 127.0.0.1:/nfstest vsit
bash: mount-t: command not found...
[root@localhost user]# mount -t nfs 127.0.0.1:/nfstest vsit
[root@localhost user]#
```

