

Submitted in partial fulfillment of the  
Requirements for the award of the Degree of

**MASTER OF SCIENCE (INFORMATION TECHNOLOGY)**

By

**SHAIKH ABDULSALAM**

**Seat Number: (3254635)**

**SUBJECT NAME:  
CLOUD COMPUTING AND SOFT COMPUTING TECHNIQUES**



*DEPARTMENT OF INFORMATION TECHNOLOGY*

**N. G. ACHARYA & D. K. MARATHE COLLEGE**  
**(Affiliated to University of Mumbai)**  
**MUMBAI , 400071**  
**MAHARASHTRA**  
**2022-23**

DEPARTMENT OF INFORMATION TECHNOLOGY

**N. G. ACHARYA & D. K. MARATHE COLLEGE**

*(Affiliated to University of Mumbai)*

MUMBAI – MAHARASHTRA - 400071

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that Shaikh Abdulsalam bearing Seat No: 3254635 submitted journal of cloud computing and soft computing techque in partial fulfillment of the requirements for the award of Degree of **MASTER OF SCIENCE in INFORMATION TECHNOLOGY** from University of Mumbai.

**Internal Guide**

**Coordinator**

**External Examiner**

**Date:**

**College Seal**

SR.NO	TITLE	DATE	SIGN
1	SOCKET PROGRAMMING		
2	RMI PROGRAMMING		
3	IMPLEMENT VIRTUALIZATION USING VM ESXI SERVER		
4	MANAGING WITH VSPHERE CLIENT		
5	IMPLEMENT XEN VIRTUALIZATION		
6	MANAGING WITH XEN CENTER		
7	IMPLEMENT VMWARE HYPER V VIRTUALIZATION		
8	Implement Windows Hyper V virtualization		
9	Develop application for Microsoft Azure		
10	Develop application for Google App Engine		

### Practical No.1 :-

**Write a program for implementing client & server communication model using TCP**

**A. A client server based program using TCP to find if the number entered is prime**

**PRIME\_SERVER:**

```
import java.net.*; import java.io.*; public class Prime_Server
{
    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();
            DataOutputStreamotc=new DataOutputStream(s.getOutputStream());    int
            y=x/2;    if(x==1 | x==2 | x==3)
            {
                otc.writeUTF(x+" is Prime Number");
                System.exit(0);
            }
            for (int i=2;i<=y;i++)
            {
                if(x%i==0)
                {
                    otc.writeUTF(x+" is not a prime number");
                }
            }
            else
            {
                otc.writeUTF(x+" is prime number");
            }
        }
    }
}
```

```

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

```

```

run:
Server Started.....
BUILD SUCCESSFUL (total time: 7 seconds)
|

```

```

Prime_CLIENT import
java.net.*; import java.io.*;
import java.lang.*; public
class Prime_Client
{
    public static void main(String args[])
    {
try
{
        Socket cs=new Socket("localhost",8001);
        BufferedReader inf=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter a number");        int a=Integer.parseInt(inf.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("Error"+e);
    }
}
}

```

```
run:
Enter a number
7
7 is prime number
BUILD SUCCESSFUL (total time: 4 seconds)
|
```

## B. A CLIENT SERVER TCP BASED CHATTING APPLICATION

**Server\_Program\_Chat** import

java.io.\*; import java.net.\*; import

java.lang.\*; public class

Server\_program\_chat

{

    public static void main(String args[])

    {

try

{

ServerSocket ss=new ServerSocket(8005);

System.out.println("Server Started");

    Socket s=ss.accept();

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

DataOutputStream out =new DataOutputStream(s.getOutputStream());

DataInputStream in=new DataInputStream(s.getInputStream());

    String send,receive;

    while((receive=in.readLine())!=null)

    {

```

        if(receive.equals("STOP"))
break;
System.out.println("Client says    "+receive);
System.out.println("Server says    ");
send=br.readLine(); out.writeBytes(send+"\n");
    }
br.close(); in.close();
out.close();
s.close();
    }
catch(Exception e)
    {
System.out.println("Error "+e);
    }
}
}

```

```

run:
Server Started
Client says      Hi Server
Server says
Hello Client How are you ?
Client says      Fine ..
Server says
ok bye
Client says
Server says
|

```

### **Client\_Program\_Chat** import

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

```
java.lang.*; public class
```

```
Client_Program_Chat
```

```
{
```

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

```
    {
```

```
try
```

```
    {
```

```
        Socket cs=new Socket("localhost",8005); BufferedReaderbr =new BufferedReader(new
InputStreamReader(System.in));
```

```
DataOutputStream out =new DataOutputStream(cs.getOutputStream());
```

```

DataInputStream in=new DataInputStream(cs.getInputStream());
System.out.println("to stop chatting type STOP");    String
msg;
    while((msg=br.readLine())!=null)
    {
out.writeBytes(msg+"\n");
        if(msg.equals("STOP"))
            Break;
System.out.println("Server Says"+in.readLine());
System.out.println("Client says");
        } br.close();
in.close(); out.close();
cs.close();
    }
catch(Exception e)
{
System.out.println("Error "+e);
    }}}

```

```

run:
to stop chatting type STOP
Hi Server
Server SaysHello Client How are you ?
Client says
Fine .
Server Saysok bye
Client says

STOP
|

```

### C. A CLIENT SERVER TCP BASED DATE APPLICATION

#### DATE\_CLIENT

```

import java.io.*; import
java.net.*; import
java.util.*; public class
Date_Client
{
    public static void main( Stringargs[])

```



```

    {
try
    {
        Socket cs=new Socket(InetAddress.getLocalHost(),8004);
BufferedReaderbr=new
BufferedReader(newInputStreamReader(cs.getInputStream()));

        String UserInput;

        while((UserInput=br.readLine())!=null)
        {
System.out.println(UserInput);
        }
cs.close();
    }
catch(Exception e)
    {
System.out.println("Error"+e);
    } } }

```

```

run:
Server Started
BUILD SUCCESSFUL (total time: 3 seconds)

```

## DATE\_SERVER

```

import java.io.*; import
java.net.*; import
java.util.*; public class
date_server
{
    public static void main( Stringargs[])
    {
try
    {
ServerSocket ss=new ServerSocket(8004);
System.out.println("Server Started");

        Socket s=ss.accept();

```

```
Calendar c=new GregorianCalendar(); PrintWriter  
pw=new PrintWriter(s.getOutputStream()); pw.println(new  
Date()); pw.println(c.get(Calendar.DATE)); pw.flush();  
s.close();  
}  
catch(Exception e)  
{  
System.out.println("Error"+e);  
} } }
```

```
run:  
Mon Nov 14 08:59:28 IST 2022  
14  
BUILD SUCCESSFUL (total time: 0 seconds)  
|
```

## Practical NO. 2

**Write a program for implementing Client Server communication model using UDP.**

**A. A client server based program using UDP to find if the number entered is even or odd.**

**udpServerEO.java** import java.io.\*; import java.net.\*; public class

UDP\_Server\_EO

{

public static void main(String args[])

{

try

{

DatagramSocket ds= new DatagramSocket(2000);

byte b[] =new byte[1024];

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

String str=new

String(dp.getData(),0,dp.getLength());

System.out.println(str); int

a=Integer.parseInt(str); String s=new String();

if(a%2==0)

s="Number is even";

else

s="Number is odd";

byte b1[]=new byte[1024];

b1=s.getBytes();

DatagramPacket dp1=new

DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000); ds.send(dp1);

}

catch(Exception e)

{

e.printStackTrace();

}

}

```
}
```

## **UDP\_Client\_EO**

```
import java.io.*; import
java.net.*; public class
UDP_Client_EO
{
    public static void main(String args[])
    {
try
    {
DatagramSocket ds=new DatagramSocket(1000);
BufferedReaderbr=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter a number :");
String num=br.readLine();      byte
b[]=new byte[1024];
b=num.getBytes();
DatagramPacketdp=new
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);
ds.send(dp);

        byte b1[] =new byte[1024];

DatagramPacket dp1=new DatagramPacket(b1,b1.length); ds.receive(dp1);
        String str=new String(dp1.getData(),0,dp1.getLength());
System.out.println(str);
    }
catch(Exception e)
    {
e.printStackTrace();
    }
}
}
```



```
Projects x UDP_Server_EO.java x UDP_Client_EO.java x Output x
Ashwin_cc_PR1 (run) x Ashwin_cc_PR1 (run) #2 x

RUN:
Enter a number :
4
Number is even
BUILD SUCCESSFUL (total time: 2 seconds)
```

## B. A client server based program using UDP to find the factorial of the entered number. `udpServerFact.java`

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

```
public class UDP_SERVER_FACTORIAL
```

```
{
```

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

```
    {
```

```
try
```

```
{
```

```
DatagramSocket ds=new DatagramSocket(2000);
```

```
    byte b[]= new byte[1024];
```

```
DatagramPacketdp=new DatagramPacket(b,b.length); ds.receive(dp);
```

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

```
System.out.println(str);        int a =Integer.parseInt(str);        int
```

```
f=1,i;
```

```
    String s=new String();
```

```
for(i=1;i<=a;i++)
```

```
{
```

```
f=f*i;
```

```
}
```

```

        s=Integer.toString(f);

        String str1="The Factorial of given number is = "+f;
byte b1[]=new byte[1024];        b1=str1.getBytes();

DatagramPacket dp1=new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000); ds.send(dp1);
    }
catch(Exception e)
    {
e.printStackTrace();
    }
}
}

```

#### **UDP\_Client\_FACTORIAL**

```

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

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

DatagramSocket ds=new DatagramSocket(1000);

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

System.out.println("Enter a number :");
String num=br.readLine();        byte
b[]=new byte[1024];
b=num.getBytes();

DatagramPacketdp=new
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);

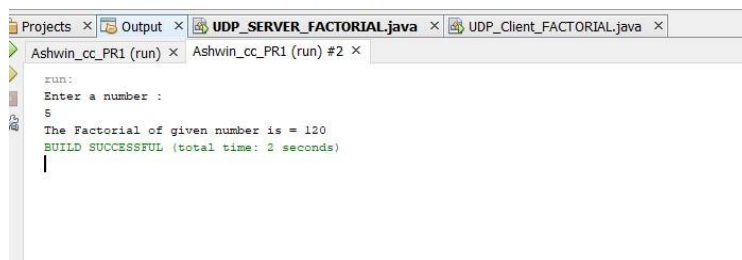
ds.send(dp);

        byte b1[] =new byte[1024];

DatagramPacket dp1=new DatagramPacket(b1,b1.length); ds.receive(dp1);

```

```
        String str=new String(dp1.getData(),0,dp1.getLength());  
        System.out.println(str);  
    }  
    catch(Exception e)  
    {  
        e.printStackTrace();  
    }  
}  
}
```



### C. A program to implement simple calculator operations like addition, subtraction, multiplication and division.

```
RPCServer.java import java.io.IOException;

import java.net.*; import java.net.*;

import java.util.*; // Main class //

Calc_Server_UDP class RPCServer { //

Main driver method public static void

main(String[] args) throws IOException

{

    // Creating a socket to listen at port 1234

    DatagramSocket ds = new DatagramSocket(1001); byte[] buf =

    null;

    // Initializing them initially with null

    DatagramPacketDpReceive = null;

    DatagramPacketDpSend = null; while

    (true) {

    buf = new byte[65535];

        // Creating a DatagramPacket to receive the data.

    DpReceive = new DatagramPacket(buf, buf.length); //

    Receiving the data in byte buffer.

    ds.receive(DpReceive);

        String inp = new String(buf, 0, buf.length);

        // Using trim() method to

        // remove extra spaces.

    inp = inp.trim();

    System.out.println("Equation Received:- " + inp);

    // Exit the server if the client sends "bye" if

    (inp.equals("bye"))

    {

    System.out.println( "Client sent bye.....EXITING");

        break;

    } int

    result;
```



```

StringTokenizer st = new StringTokenizer(inp);          int
oprnd1 = Integer.parseInt(st.nextToken());           String
operation = st.nextToken();          int oprnd2 =
Integer.parseInt(st.nextToken());          if
(operation.equals("+"))          result = oprnd1 + oprnd2;
else if (operation.equals("-"))          result = oprnd1 -
oprnd2;          else if (operation.equals("*"))          result
= oprnd1 * oprnd2;
          else
          result = oprnd1 / oprnd2;
System.out.println("Sending the result...");
          String res = Integer.toString(result);          //
Clearing the buffer after every message buf =
res.getBytes()
          // Getting the port of client
int port = DpReceive.getPort(); DpSend =
new DatagramPacket( buf, buf.length,
InetAddress.getLocalHost(),
          port);
ds.send(DpSend);
    }
}
}

RPCCLIENT import java.io.IOException;
import java.net.*; import java.net.*;
import java.util.*; // Calc_Client_UDP
public class RPCClient { // Main driver
method    public static void main(String
args[])    throws IOException
    {
        Scanner sc = new Scanner(System.in);

```

```

DatagramSocket ds = new DatagramSocket();
InetAddress ip = InetAddress.getLocalHost();
byte buf[] = null;    while (true) {
System.out.print("Enter the equation in the format:");
System.out.println( "operand1 operator operand2");

        // Awaiting from entered input
String inp = sc.nextLine(); buf = new
byte[65535];

        // Converting the String input into the byte
        // array
buf = inp.getBytes();

        // Step 2
        // Creating the datagramPacket for sending the
        // data.
DatagramPacket DpSend = new DatagramPacket( buf,
buf.length, ip, 1001);

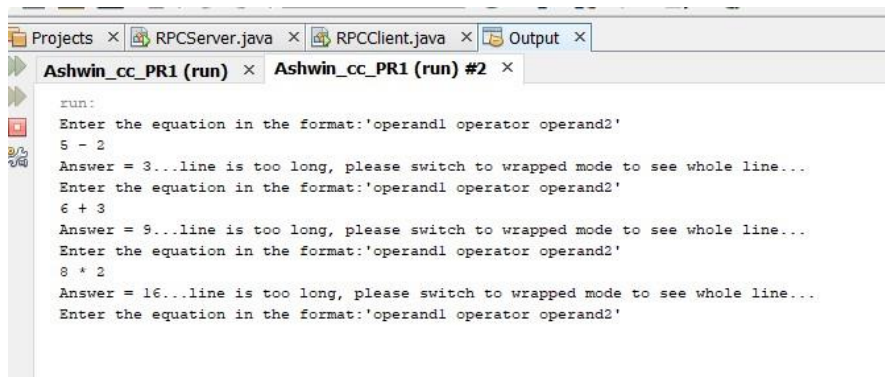
        // Invoking the send call to actually send the    //
data.
ds.send(DpSend);

        // Break the loop if user enters "bye"
// using the break keyword    if
(inp.equals("bye"))    break;
buf = new byte[65535];

DatagramPacket DpReceive = new DatagramPacket(buf, buf.length); ds.receive(DpReceive);
System.out.println("Answer = " + new String(buf, 0, buf.length));

    }
}
}

```

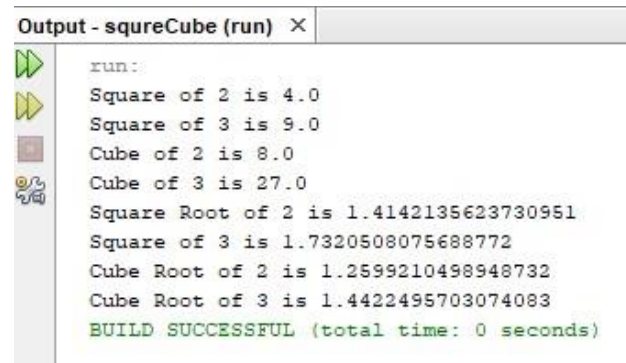


```
Projects x RPCServer.java x RPCClient.java x Output x
Ashwin_cc_PR1 (run) x Ashwin_cc_PR1 (run) #2 x

run:
Enter the equation in the format:'operand1 operator operand2'
5 - 2
Answer = 3...line is too long, please switch to wrapped mode to see whole line...
Enter the equation in the format:'operand1 operator operand2'
6 + 3
Answer = 9...line is too long, please switch to wrapped mode to see whole line...
Enter the equation in the format:'operand1 operator operand2'
8 * 2
Answer = 16...line is too long, please switch to wrapped mode to see whole line...
Enter the equation in the format:'operand1 operator operand2'
```

#### D. A program that finds the square, square root, cube and cube root of the entered

```
package squrecube; public
class SquireCube {
    public static void main(String[] args) {
        int num1=2,num2=3;
        System.out.println("Square of "+num1+" is "+Math.pow(num1,2));
        System.out.println("Square of "+num2+" is "+Math.pow(num2,2));
        System.out.println("Cube of "+num1+" is "+Math.pow(num1,3));
        System.out.println("Cube of "+num2+" is "+Math.pow(num2,3));
        System.out.println("Square Root of "+num1+" is "+Math.sqrt(num1));
        System.out.println("Square of "+num2+" is "+Math.sqrt(num2));
        System.out.println("Cube Root of "+num1+" is "+Math.cbrt(num1));
        System.out.println("Cube Root of "+num2+" is "+Math.cbrt(num2));
    }
}
```



```
Output - squreCube (run) x

run:
Square of 2 is 4.0
Square of 3 is 9.0
Cube of 2 is 8.0
Cube of 3 is 27.0
Square Root of 2 is 1.4142135623730951
Square of 3 is 1.7320508075688772
Cube Root of 2 is 1.2599210498948732
Cube Root of 3 is 1.4422495703074083
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Practical No: 03

### A multicast Socket example.

```
BroadcastServer.java import
java.net.*; import java.io.*; import
java.util.*; public class BroadcastServer
{ public static final int PORT = 1234;
public static void main(String args[])throws Exception {
MulticastSocket socket;
DatagramPacket packet;
InetAddress address;

// set the multicast address to your local subnet address =
InetAddress.getByName("239.1.2.3"); socket = new MulticastSocket();

// join a Multicast group and send the group messages socket.joinGroup(address); byte[] data =
null; for(;;)
{
Thread.sleep(10000);
System.out.println("Sending "); String str = ("This is Neha Calling...."); data
= str.getBytes();
packet = new DatagramPacket(data, str.length(),address,PORT);
// Sends the packet socket.send(packet);
} // end for
} // end main
} // end class BroadcastServer

BroadcastClient.java import
java.net.*; import java.io.*;

public class BroadcastClient { public
static final int PORT = 1234;
public static void main(String args[])throws Exception {
MulticastSocket socket;
DatagramPacket packet;
InetAddress address;

// set the mulitcast address to your local subnet address =
```

```

InetAddress.getByName("239.1.2.3"); socket = new MulticastSocket(PORT);

//join a Multicast group and wait for a message socket.joinGroup(address); byte[] data = new
byte[100];

packet = new DatagramPacket(data,data.length); for(;;)
{
// receive the packets socket.receive(packet);

String str = new String(packet.getData()); System.out.println("Message received from "+
packet.getAddress() + "

Message is : "+str);

} // for
} // main
} // end BroadcastClient Output:

```

 C:\WINDOWS\system32\cmd.exe

E:\Ds\_Yugi>javac BroadcastServer.java

E:\Ds\_Yugi>java BroadcastServer  
Sending  
Sending  
Sending  
- ..

E:\Ds\_Yugi>javac BroadcastClient.java

E:\Ds\_Yugi>java BroadcastClient

Message received from /10.29.26.232 Message is: This is Neha Calling....

Message received from /10.29.26.232 Message is: This is Neha Calling....

Message received from /10.29.26.232 Message is: This is Neha Calling....

Message received from /10.29.26.232 Message is: This is Neha Calling....

## **Practical No: 04**

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

**A. A RMI based application program to display current date and time.**

#### **InterDate.java**

```
import java.rmi.*;

public interface InterDate extends Remote
{ public String display() throws Exception;
}
```

#### **ServerDate.java**

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

public class ServerDate extends UnicastRemoteObject implements InterDate {
public ServerDate() throws Exception
{
}

public String display() throws Exception
{
String str = "";
Date d = new Date(); str = d.toString(); return str;
}

public static void main(String args[]) throws Exception {
ServerDate s1 = new ServerDate();
Naming.bind("DS",s1);
System.out.println("Object registered.....");

}
}
```

#### **ClientDate.java**

```
import java.rmi.*; import
java.io.*; public class
ClientDate
{
```

```

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

String s1;

InterDate h1 = (InterDate)Naming.lookup("DS"); s1 = h1.display(); System.out.println(s1);

}

}

```

```

C:\WINDOWS\system32\cmd.exe - rmdir
E:\Ds_Yugi>javac ServerDate.java
E:\Ds_Yugi>javac ClientDate.java
E:\Ds_Yugi>rmic ServerDate
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
E:\Ds_Yugi>rmdir

```

```

C:\WINDOWS\system32\cmd.exe - java ServerDate
E:\Ds_Yugi>java ServerDate
Object registered.....

```

```

C:\WINDOWS\system32\cmd.exe
E:\Ds_Yugi>java ClientDate
Thu Jan 04 17:38:00 IST 2018

```

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

**1. InterConvert.java** import

```

java.rmi.*;

public interface InterConvert extends Remote

{ public String convertDigit(String no) throws Exception; }

```

**2. ServerConvert.java**

```

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

public class ServerConvert extends UnicastRemoteObject implements InterConvert {

public ServerConvert() throws Exception

{

}

}

```

```

public String convertDigit(String no) throws Exception { String str = "";
for(int i = 0; i < no.length(); i++)
{ int p = no.charAt(i); if( p == 48)
{ str += "zero ";
} if( p == 49)
{ str += "one ";
} if( p == 50)
{ str += "two ";
} if( p == 51)
{ str += "three ";
} if( p == 52)
{ str += "four ";

} if( p == 53)
{ str += "five ";
} if( p == 54)
{ str += "six ";
} if( p == 55)
{
str += "seven ";
}
if( p == 56)
{
str += "eight ";
}
if( p == 57)
{ str += "nine ";
}
} return str;
}

public static void main(String args[]) throws Exception {
ServerConvert s1 = new ServerConvert();

```



```

Naming.bind("Wrd",s1);

System.out.println("Object registered...."); }

}

```

### 3. ClientConvert.java import

```

java.rmi.*;

import java.io.*; public class ClientConvert

{

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

        InterConvert h1 =

        (InterConvert)Naming.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);

    }

}

```

```

C:\WINDOWS\system32\cmd.exe - rmiregistry
E:\Ds_Yugi>javac ServerConvert.java
E:\Ds_Yugi>javac ClientConvert.java
E:\Ds_Yugi>rmic ServerConvert
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.
E:\Ds_Yugi>rmiregistry

C:\WINDOWS\system32\cmd.exe - java ServerConvert
E:\Ds_Yugi>java ServerConvert
Object registered....

C:\WINDOWS\system32\cmd.exe
E:\Ds_Yugi>java ClientConvert
Enter a number :
123
The word representation of the entered digit is : one two three
E:\Ds_Yugi>java ClientConvert
Enter a number :
456
The word representation of the entered digit is : four five six

```

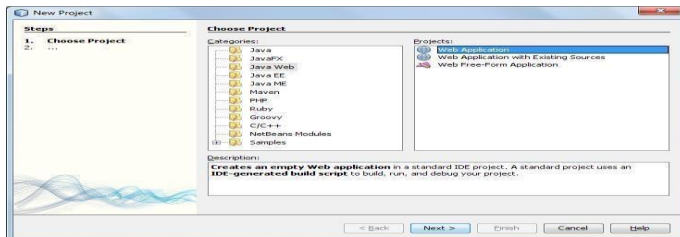
## Practical 5

### Implementing “Big” Web Service.

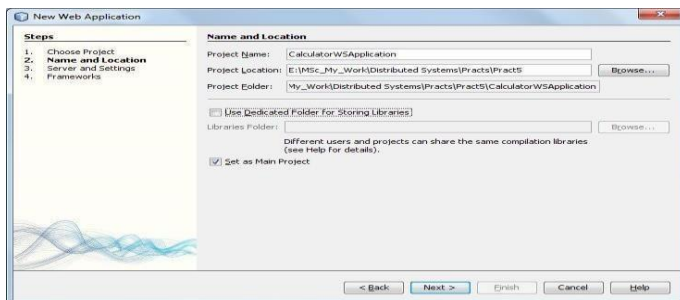
#### 1. Creating a Web Service

##### A. Choosing a Container:

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



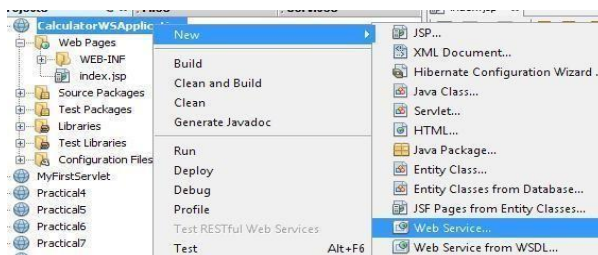
2. Name the project CalculatorWSApplication. Select a location for the project. Click Next.



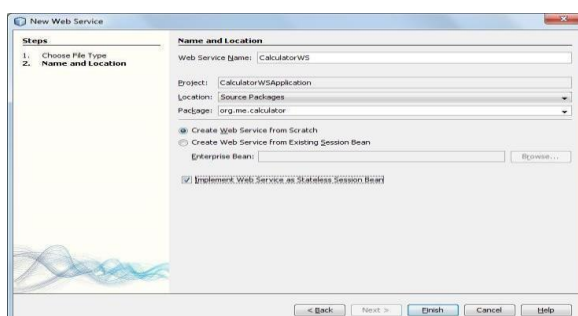
3. Select your server and Java EE version and click Finish.

##### 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.



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

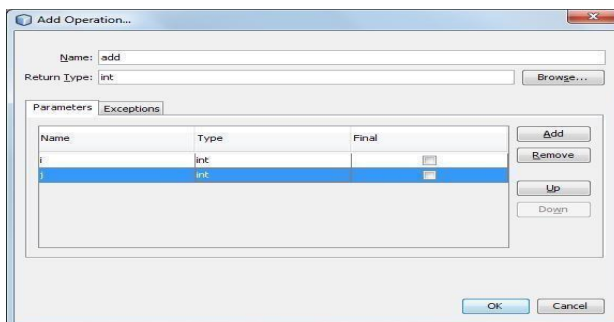
## 2. Adding an Operation to the Web Service

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

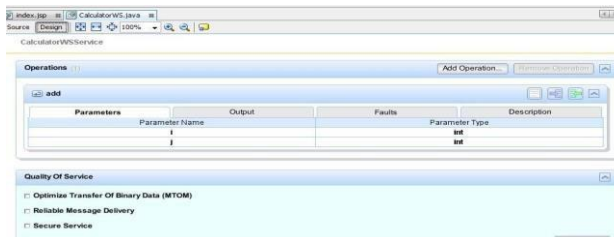
1. Change to the Design view in the editor.



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:



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:



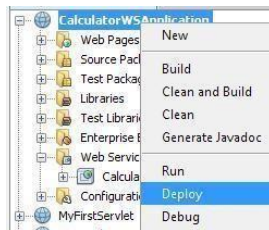
8. Click Source. And code the following.

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

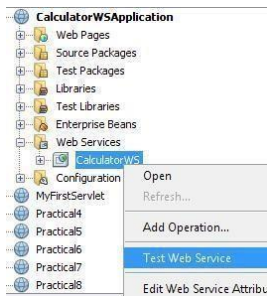
## 3. Deploying and Testing the Web Service

### 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.



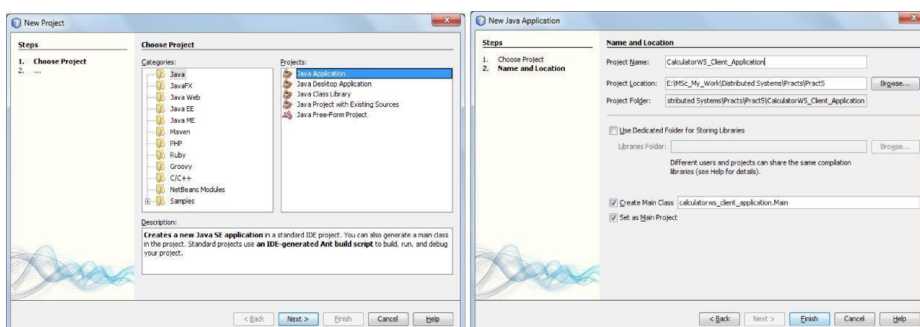
3. The IDE opens the tester page in your browser, if you deployed a web application to the GlassFish server.
4. If you deployed to the GlassFish server, type two numbers in the tester page, as shown below:



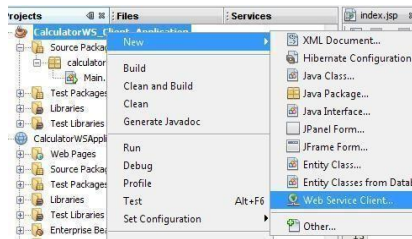
5. The sum of the two numbers is displayed:
4. Consuming the Web Service

#### A. 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.\



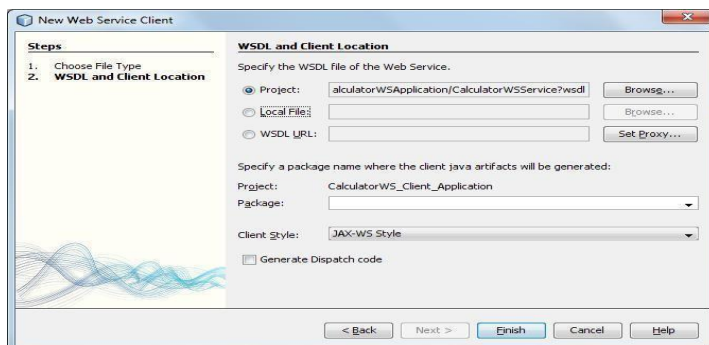
2. Right-click the CalculatorWS\_Client\_Application node and choose New > Web Service Client. The New Web Service Client wizard opens.



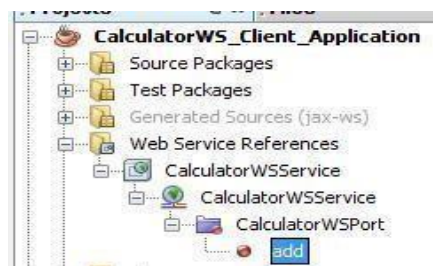
3. Select Project as the WSDL source. Click Browse. Browse to the CalculatorWS web service in the CalculatorWSApplication project. When you have selected the web service, click OK.



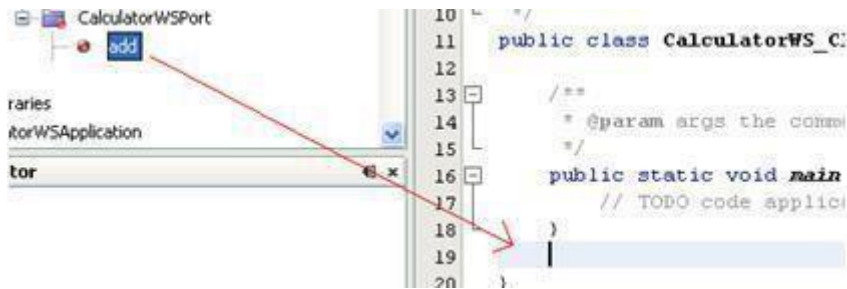
4. Do not select a package name. Leave this field empty.



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.

```
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); }
```

```
}
```

9. 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)

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

### **A. 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");

## **B. Creating a web service**

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

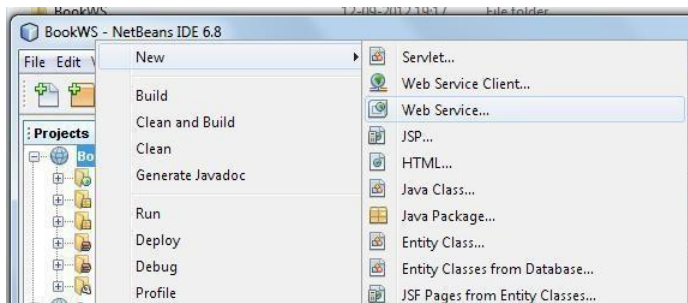
- Web service can be either deployed in a Web container or in an EJB container.
- If a Java EE 6 application is created, use a Web container because EJBs can be placed directly in a Web application.

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

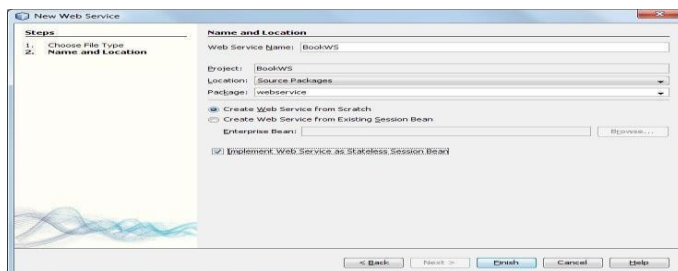
- To create a Web application, select File - New Project.
- New Project dialog box appears. Select Java Web available under the Categories section and Web Application available under the Projects section. Click Next.
- New Web Application dialog box appears. Enter BookWS as the project name in the Project 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.



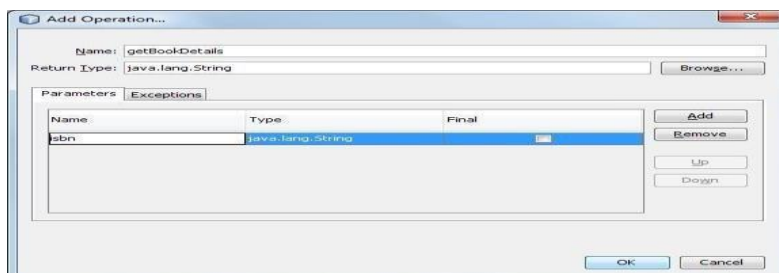
- Click Finish.
- The web service in the form of java class is ready.

#### 4. Designing the web service

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

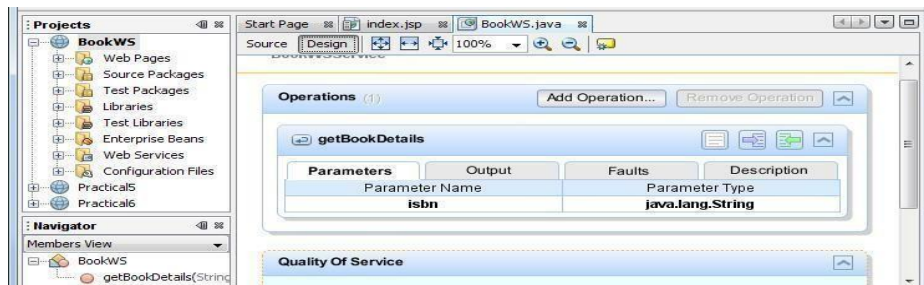
##### i. Adding an operation to the web service

- 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.

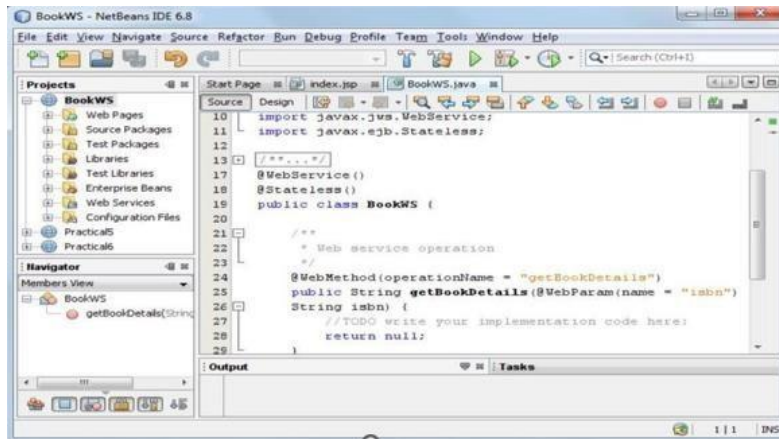


- Click Ok. The design view displays the operation added as shown in the diagram.





- 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;

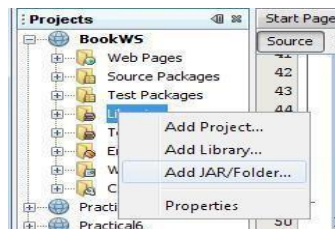
}

}

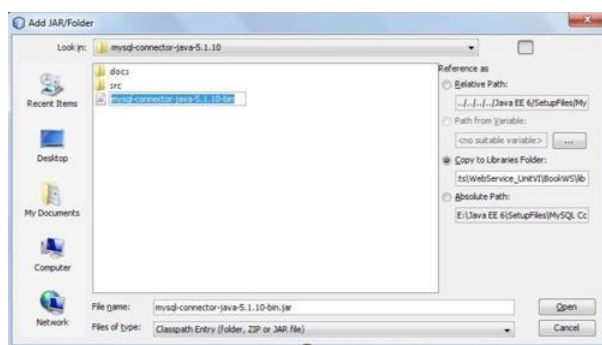
```

## 5. 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.



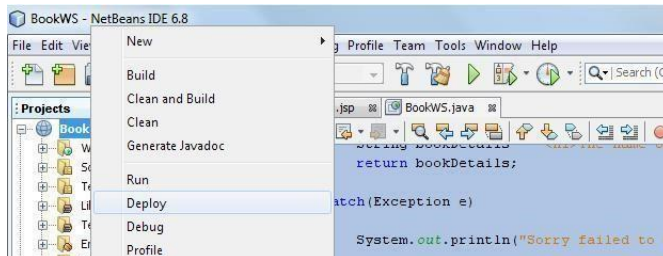
- Choose the location where mysql-connector-java-5.1.10-bin is located, select it and click on open as shown



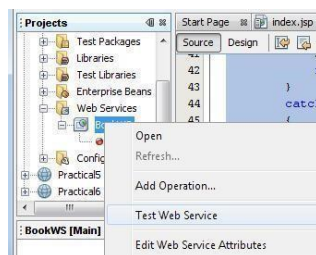
## 6. Deploying and testing the web service

- 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.



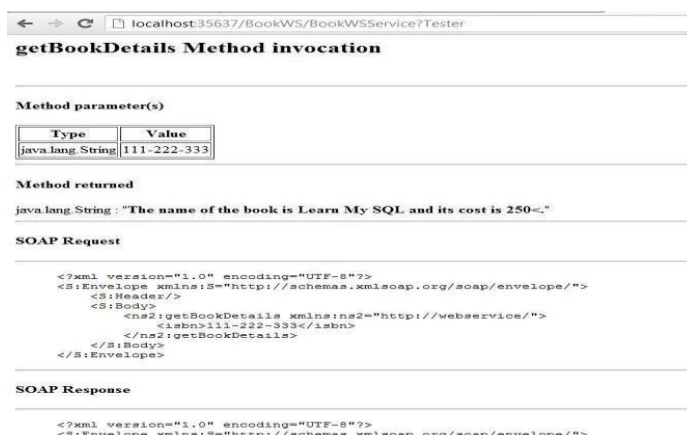
- The IDE starts the server, builds the application and deploys the application to the server.
- Follow the progress of these operations in the BookWS (run-deploy) and GlassFish v3 Domain tabs in the Output view.
- Now expand the web services directory of the BookWS project, right-click the BookWS Web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



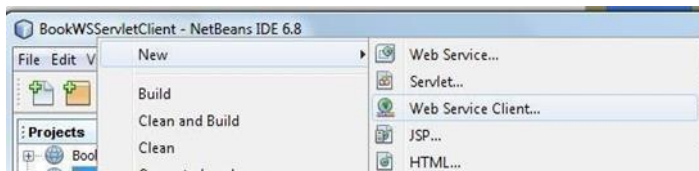
## 7. Consuming the web service

### i. Creating a web application

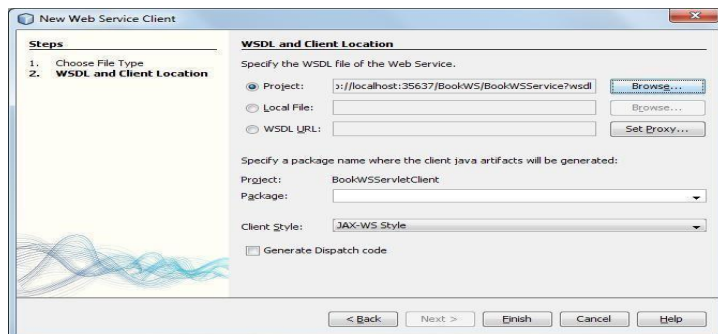
- To create a web application, select File -> New Project.
- New project dialog box appears, select java web available under the categories section and web application available under the projects section. Click Finish.
- New web application dialog box appears. Enter BookWSServletClient as the project 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

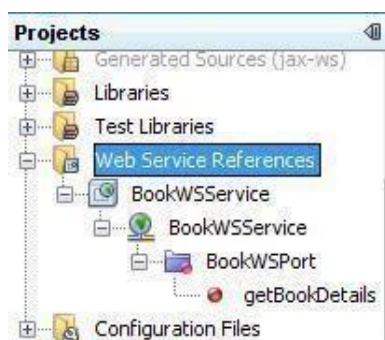
- Right-click the BookWSServletClient project and select New -> Web Service Client as shown in the diagram.



- 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.



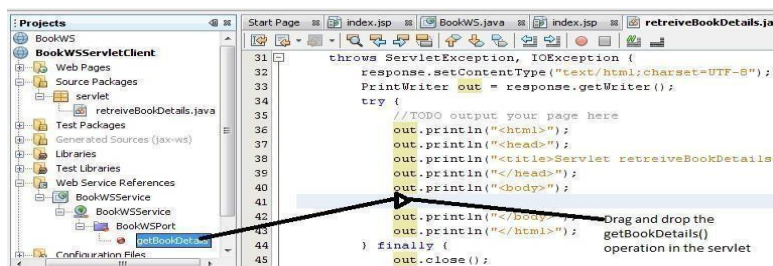
- Leave the other settings as it is. Click Finish
- The Web Service Reference directory is added to the BookWSServletClient application as shown in the diagram. It displays the structure of the newly created client including the getBookDetails() method created earlier.



### 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

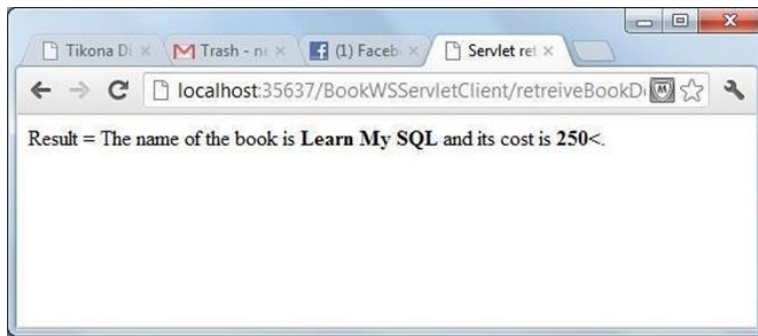
- Build the web application.
- Right click BookWSServletClient project and select Build.
- Once the Build menu item is clicked the details about the compilation and building of the BookWSServletClient Web application appears in the output – BookWSServletClient (dist) window.

#### 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.





## Practical: 06

### Implement Xen virtualization and manage with Xen Center

- Install XenServer in VMware Workstation and select Guest operating system as Linux.

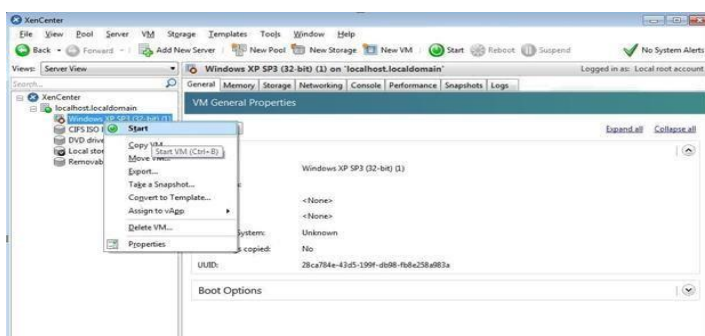


Note IP Address – “ 192.168.124.137” ping it from command prompt.

- Now Install Citrix App if not installed
- Now Open Citrix XenCenter – and Click and Add Server.

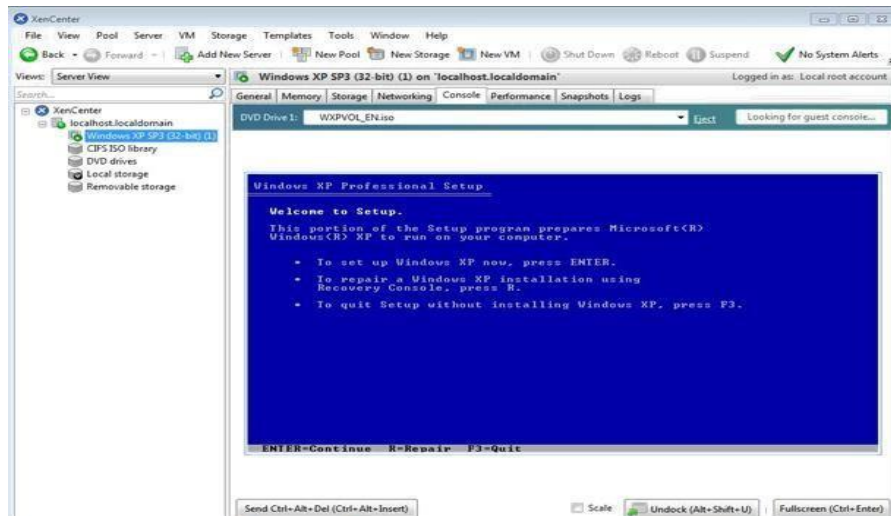


- Fill IP address copied from Installation and User name as “root” and Password as “root123” which we had given during installation and Click on Add.
- Then click on Ok
- Now Click on New Storage
- Select Window File Sharing (CIFS) and click on next • Uncheck Auto generate option Click on Next.
- Provide the path of shared windows XP image and enter local pc credential , click on Finish
- Click on New VM – and Windows XP SP3
- Select ISO file and click on next – • Click on Next – ○ Uncheck – Start the new VM and click on create now
- Now Right click on Windows XP and Start –





- Installation is successful and virtual node has been created if we get below Welcome screen of Windows XP machine.

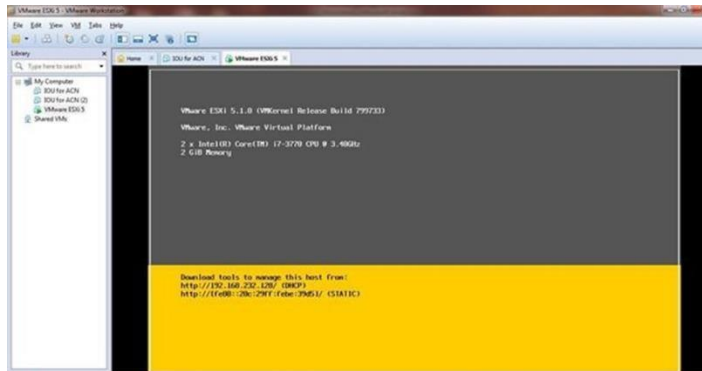


## Practical: 07

### Implement virtualization using VMWare ESXi Server and managing with vCenter

#### Steps:

1. Install ESXi iso in VMWare workstation.



2. Install VMware vSphere Client



3. In vSphere create new Virtual Machine. Install Windows XP iso file and open it.



## Practical: 08

### Implement Windows Hyper V virtualization

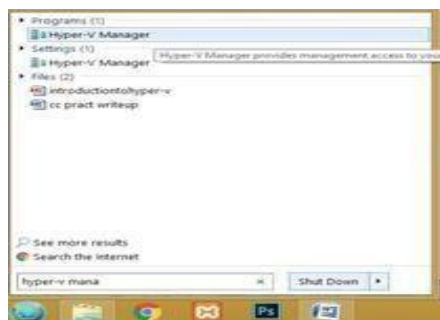
- First we have to uninstall vmware software if already installed on computer because the VMware Workstation installer does not support running on a HyperV virtual machine. after uninstalling vmware we can proceed to next step go to control panel and click on uninstall a program.



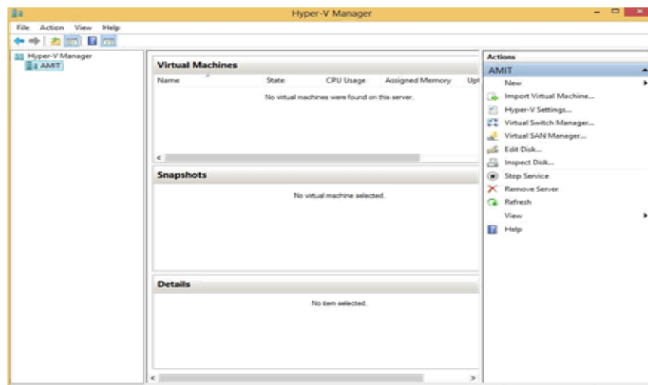
- Click on Turn windows features on or off.
- Now in windows features check on Hyper-V option.



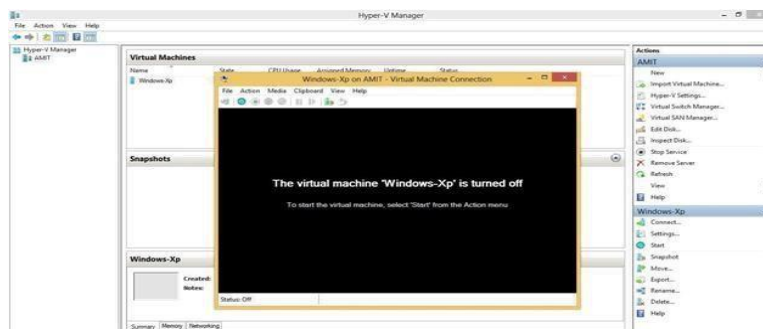
- After Restart Search for hyper-v manager in search box and click on that.



- for creating virtual machine first we have to create virtual switch click on virtual switch manager option.



- Select External as a connection type and then click on create virtual switch.
- Create new Virtual switch and install windows XP .iso and virtual machine will start.



## Practical: 09

### Develop application for Microsoft Azure.

#### **Step 1:**

To develop an application for Windows Azure on Visual Studio install the “Microsoft Azure SDK for .NET (VS 2010) – 2.8.2.1”

#### **Step2:**

Turn windows Features ON or OFF:

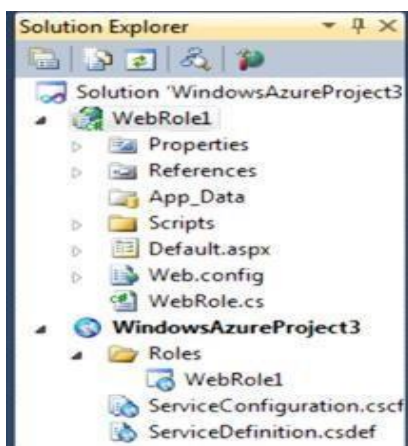
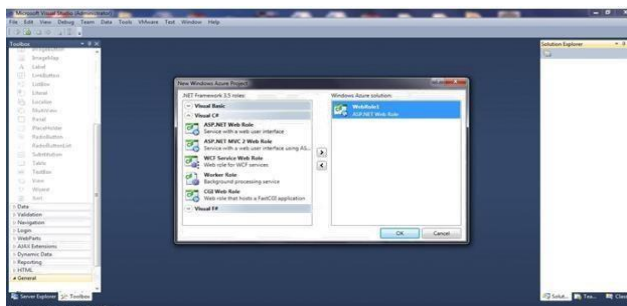
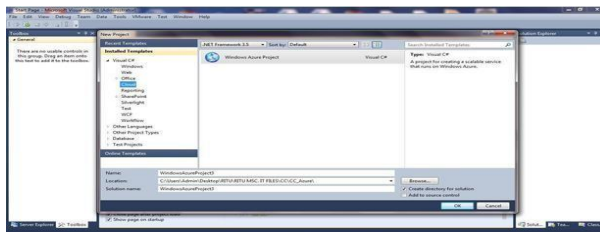
Go to Control panel and click on programs.

Turn Windows features on or off.

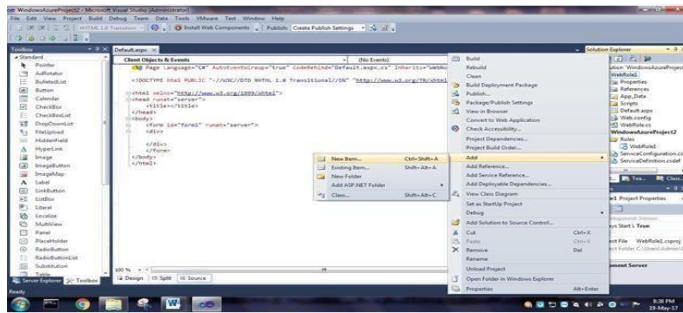
#### **Step3 :**

Now, Start the visual studio 2010 and Go To File->New->Project

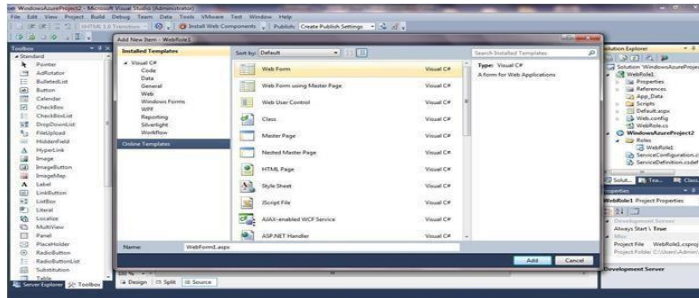
Expand Visual C#-> Select Cloud



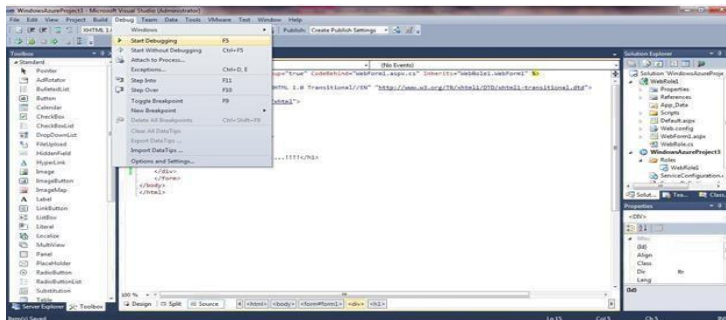
- Right Click on WebRole1>>ADD>>New Item



○ Add a New web Form. Give it a name. Click Add



○ Deploy the project:



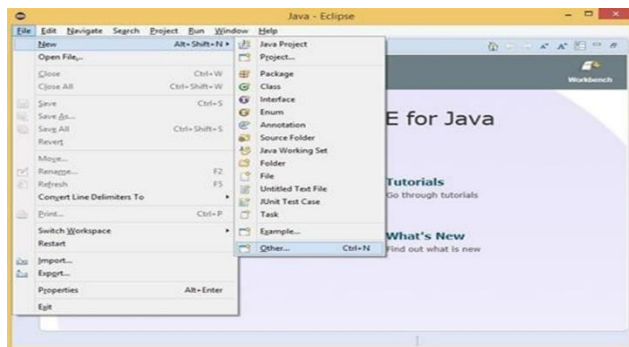
○ Run Project



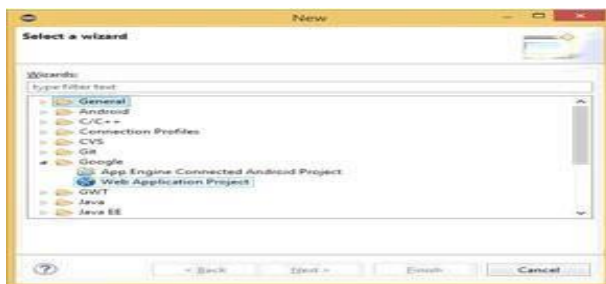
## Practical: 10

### Develop application for Google App Engine

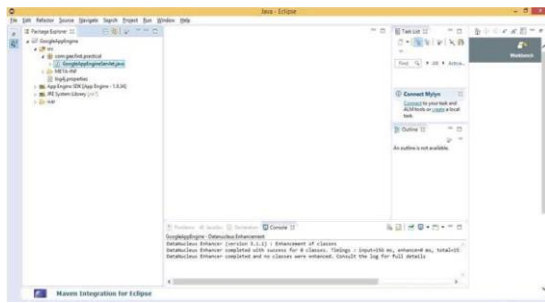
- Open Eclipse Luna. Go to Help Menu Install New Software...
- In Install window Click on the “Add” button besides the Work with textbox. Add Repository window appears. Enter the Location as “https://dl.google.com/eclipse/plugin/4.4” and click on “OK” button.
- From the available softwares select the required softwares and tools as shown in the below image for the GAE. Then click on the “Next” button.
- In the Install Details window click on “Next” button.
- In the Next Window "Review the Items to be Installed" then click on “Next”
- In the next window for Review Licenses select the option “I accept.....” and click on “Finish” • button.
- After Installation you will get option to "Restart Eclipse", click on Yes. So that the software you selected gets updated... ○ Now, go to File Menu\_New\_Other.



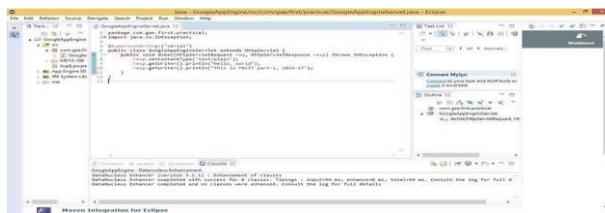
- In the New window select Google\_Web Application Project and click on “Next” button.



- Enter the details for the new Web application project. Deselect the Use Google Web Toolkit option under the section Google SDKs. Click on the “Finish” button.
- From the Package Explorer open the .java file (Here it “Google\_App\_EngineServlet.java”).



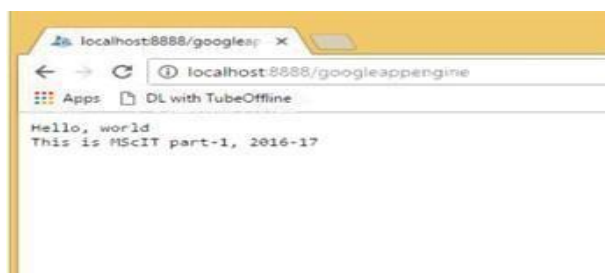
- Edit the file as required (Unedited file too can be used. Here the editing is done to “what should be displayed” on the browser). Save the file. Click on the Run option available on the Tools bar



- In the browser (Here, Google Chrome) type the address as “localhost:8888” which is "Default".



- In localhost:8888 the link to the Google\_App\_EngineServlet.java file as Google\_App\_Engine is displayed. Click on this link. It will direct you to “localhost:8888/Google\_App\_Engine”.



- The output text entered in the java program is displayed as the output when clicked the link “Google\_App\_Engine”.



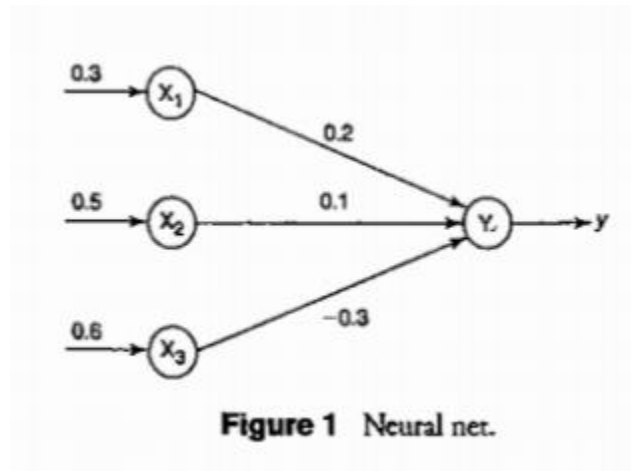
## **INDEX**

Sr. No.	Practicals	Signature
1.	Calculate the output of neural net using both binary and bipolar sigmoidal function.	
2.	Calculate the net input for the network shown in Figure 2 with bias included in the network.	
3.	Generate AND/NOT function using McCulloch-Pitts neural net.	
4.	Generate XOR function using McCulloch-Pitts neural net.	
5.	Write a program for linear separation.	
6.	Write a program to implement membership and identity operators is, is not.	
7.	Write a program to implement membership and identity operators in, not in.	

## PRACTICAL NO. : 1

**AIM:** Calculate the output of neural net using both binary and bipolar sigmoidal function.

For the network shown in the figure 1, calculate the net input to output neuron.



**SOLUTION:** The given neural net consist of three input neurons and one output neuron. The inputs and weight are

$$[x_1, x_2, x_3] = [0.3, 0.5, 0.6]$$

$$[w_1, w_2, w_3] = [0.2, 0.1, -0.3]$$

The net input can be calculated as

$$\begin{aligned} Y_{in} &= x_1w_1 + x_2w_2 + x_3w_3 \\ &= 0.3*0.2 + 0.5*0.1 + 0.6*(-0.3) \\ &= -0.07 \end{aligned}$$

## CODE:

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:

# number of elements as input
```

```

n = int(input("Enter number of elements : "))

# In[2]:
print("Enter the inputs")
inputs = [] # creating an empty list for inputs

# iterating till the range
for i in range(0, n):
    ele = float(input())

    inputs.append(ele) # adding the element

print(inputs)

# In[3]:
print("Enter the weights")
# creating an empty list for weights
weights = []

# iterating till the range
for i in range(0, n):
    ele = float(input())
    weights.append(ele) # adding the element

print(weights)

# In[4]:
print("The net input can be calculated as  $Y_{in} = x_1w_1 + x_2w_2 + x_3w_3$ ")

# In[5]:
Yin = []
for i in range(0, n):
    Yin.append(inputs[i]*weights[i])
print(round(sum(Yin),3))

```

## Output :

```
In [1]: # number of elements as input
n = int(input("Enter number of elements : "))
Enter number of elements : 3
```

```
In [2]: print("Enter the inputs")
inputs = [] # creating an empty list for inputs

# iterating till the range
for i in range(0, n):
    ele = float(input())

    inputs.append(ele) # adding the element
print(inputs)

Enter the inputs
0.3
0.5
0.6
[0.3, 0.5, 0.6]
```

```
In [3]: print("Enter the weights")
# creating an empty list for weights
weights = []

# iterating till the range
for i in range(0, n):
    ele = float(input())

    weights.append(ele) # adding the element
print(weights)

Enter the weights
0.2
0.1
-0.3
[0.2, 0.1, -0.3]
```

```
In [4]: print("The net input can be calculated as   Yin = x1w1 + x2w2 + x3w3")

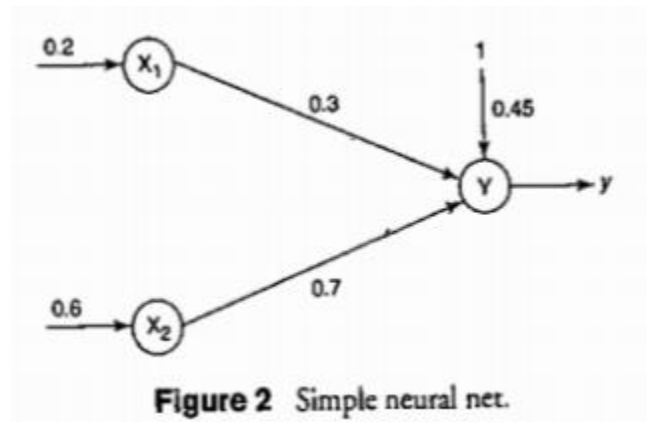
The net input can be calculated as   Yin = x1w1 + x2w2 + x3w3
```

```
In [5]: Yin = []
for i in range(0, n):
    Yin.append(inputs[i]*weights[i])
print(round(sum(Yin),3))

-0.07
```

## PRACTICAL NO. : 2

**AIM:** Calculate the net input for the network shown in Figure 2 with bias included in the network.



**SOLUTION:** The given net consists of two input neurons, a bias and an output neuron. The inputs are

$[x_1, x_2] = [0.2, 0.6]$  and the weights are  $[w_1, w_2] = [0.3, 0.7]$ . Since the bias is included  $b = 0.45$  and bias input  $x_0$  is equal to 1, the net input is calculated as

$$\begin{aligned} Y_{in} &= b + x_1 w_1 + x_2 w_2 \\ &= 0.45 + 0.2 \times 0.3 + 0.6 \times 0.7 \\ &= 0.45 + 0.06 + 0.42 = 0.93 \end{aligned}$$

Therefore  $y_m = 0.93$  is the net input.

### Code :

```
#!/usr/bin/env python
# coding: utf-8
# In[1]:

# number of elements as input
n = int(input("Enter number of elements : "))
```

```
# In[2]:
print("Enter the inputs")
inputs = [] # creating an empty list for inputs

# iterating till the range
for i in range(0, n):
    ele = float(input())

    inputs.append(ele) # adding the element

print(inputs)
```

```
# In[3]:
print("Enter the weights")
# creating an empty list for weights
weights = []

# iterating till the range
for i in range(0, n):
    ele = float(input())
    weights.append(ele) # adding the element

print(weights)
```

```
# In[4]:
print("The net input can be calculated as  $Y_{in} = b + x_1w_1 + x_2w_2$ ")
```

```
# In[5]:
b=0.45;
Yin = []
for i in range(0, n):
    Yin.append(inputs[i]*weights[i])
Print(round((sum(Yin)+b),3))
```

## OUTPUT:

```
In [1]: # number of elements as input
n = int(input("Enter number of elements : "))
Enter number of elements : 2
```

```
In [2]: print("Enter the inputs")
inputs = [] # creating an empty list for inputs

# iterating till the range
for i in range(0, n):
    ele = float(input())
    inputs.append(ele) # adding the element
print(inputs)

Enter the inputs
0.2
0.6
[0.2, 0.6]
```

```
In [3]: print("Enter the weights")
# creating an empty list for weights
weights = []

# iterating till the range
for i in range(0, n):
    ele = float(input())
    weights.append(ele) # adding the element
print(weights)

Enter the weights
0.3
0.7
[0.3, 0.7]
```

```
In [4]: print("The net input can be calculated as   Yin = b + x1w1 + x2w2")

The net input can be calculated as   Yin = b + x1w1 + x2w2
```

```
In [8]: # value of bias as input
b = float(input("Enter number of Bias : "))
Enter number of Bias : 0.45
```

```
In [10]: Yin = []
for i in range(0, n):
    Yin.append(inputs[i]*weights[i])
round((sum(Yin)+b),3)
```

```
Out[10]: 0.93
```

## PRACTICAL NO: 03

**AIM:** Generate AND/NOT function using McCulloch-Pitts neural net.

### **CODE:**

```
import numpy
# enter the no of inputs
num_ip = int(input("Enter the number of inputs : "))
#Set the weights with value 1
w1 = 1
w2 = 1
print("For the ", num_ip , " inputs calculate the net input using  $y_{in} = x_1w_1 + x_2w_2$ 
")
x1 = []
x2 = []
for j in range(0, num_ip):
    ele1 = int(input("x1 = "))
    ele2 = int(input("x2 = "))
    x1.append(ele1)
    x2.append(ele2)

print("x1 = ",x1)
print("x2 = ",x2)
n = x1 * w1
m = x2 * w2
Yin = []
for i in range(0, num_ip):
    Yin.append(n[i] + m[i])
print("Yin = ",Yin)
#Assume one weight as excitatory and the other as inhibitory, i.e.,
n1 = x1 * w1
m1 = x2 * w2

Yin = []
for i in range(0, num_ip):
    Yin.append(n1[i] - m1[i])
```



```

print("After assuming one weight as excitatory and the other as inhibitory Yin =
",Yin)
#From the calculated net inputs, now it is possible to fire the neuron for input (1,
0)
#only by fixing a threshold of 1, i.e.,  $\theta \geq 1$  for Y unit.
#Thus,  $w_1 = 1$ ,  $w_2 = -1$ ;  $\theta \geq 1$ 
Y=[]
for i in range(0, num_ip):
    if(Yin[i]>=1):
        ele = 1
        Y.append(ele)
    if(Yin[i]<1):
        ele = 0
        Y.append(ele)
print("Y = ",Y)

```

## **OUTPUT:**

```

Enter the number of inputs : 4
For the 4 inputs calculate the net input using yin = x1w1 + x2w2

x1 = 0
x2 = 0
x1 = 0
x2 = 1
x1 = 1
x2 = 0
x1 = 1
x2 = 1
x1 = [0, 0, 1, 1]
x2 = [0, 1, 0, 1]
Yin = [0, 1, 1, 2]
After assuming one weight as excitatory and the other as inhibitory Yin = [0, 1, -1, 0]
Y = [0, 1, 0, 0]

In [14]: |

```

## **PRACTICAL NO. : 4**

**AIM:** Generate XOR function using McCulloch-Pitts neural net.

### **CODE:**

```
import numpy
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt

def tanh(x):
    return (1.0 - numpy.exp(-2*x))/(1.0 + numpy.exp(-2*x))

def tanh_derivative(x):
    return (1 + x)*(1 - x)
class NeuralNetwork:
    def __init__(self, net_arch):
        numpy.random.seed(0)
# Initialized the weights, making sure we also
# initialize the weights for the biases that we will add later
        self.activity = tanh
        self.activity_derivative = tanh_derivative
        self.layers = len(net_arch)
        self.steps_per_epoch = 1
        self.arch = net_arch
        self.weights = []
# Random initialization with range of weight values (-1,1)
        for layer in range(self.layers - 1):
            w = 2*numpy.random.rand(net_arch[layer] + 1, net_arch[layer+1]) - 1
            self.weights.append(w)
    def _forward_prop(self, x):
        y = x
        for i in range(len(self.weights)-1):
            activation = numpy.dot(y[i], self.weights[i])
            activity = self.activity(activation)
# add the bias for the next layer
            activity = numpy.concatenate((numpy.ones(1), numpy.array(activity)))
            y.append(activity)
```

```

# last layer
    activation = numpy.dot(y[-1], self.weights[-1])
    activity = self.activity(activation)
    y.append(activity)
    return y
def _back_prop(self, y, target, learning_rate):
    error = target - y[-1]
    delta_vec = [error * self.activity_derivative(y[-1])]
# we need to begin from the back, from the next to last layer
    for i in range(self.layers-2, 0, -1):
        error = delta_vec[-1].dot(self.weights[i][1:].T)
        error = error*self.activity_derivative(y[i][1:])
        delta_vec.append(error)
# Now we need to set the values from back to front
    delta_vec.reverse()
# Finally, we adjust the weights, using the backpropagation rules
    for i in range(len(self.weights)):
        layer = y[i].reshape(1, self.arch[i]+1)
        delta = delta_vec[i].reshape(1, self.arch[i+1])
        self.weights[i] += learning_rate*layer.T.dot(delta)
#####
# parameters
# -----
# self: the class object itself
# data: the set of all possible pairs of booleans True or False indicated by the
integers 1 or 0
# labels: the result of the logical operation 'xor' on each of those input pairs
#####
    def fit(self, data, labels, learning_rate=0.1, epochs=100):
# Add bias units to the input layer -
# add a "1" to the input data (the always-on bias neuron)
        ones = numpy.ones((1, data.shape[0]))
        Z = numpy.concatenate((ones.T, data), axis=1)
        for k in range(epochs):
            if (k+1) % 10000 == 0:
                print('epochs: {}'.format(k+1))
                sample = numpy.random.randint(X.shape[0])

```

```

# We will now go ahead and set up our feed-forward propagation:
    x = [Z[sample]]
    y = self._forward_prop(x)
# Now we do our back-propagation of the error to adjust the weights:
    target = labels[sample]
    self._back_prop(y, target, learning_rate)
#####
# the predict function is used to check the prediction result of
# this neural network.
#
# parameters
# -----
# self: the class object itself
# x: single input data
#####
    def predict_single_data(self, x):
        val = numpy.concatenate((numpy.ones(1).T, numpy.array(x)))
        for i in range(0, len(self.weights)):
            val = self.activity(numpy.dot(val, self.weights[i]))
            val = numpy.concatenate((numpy.ones(1).T, numpy.array(val)))
        return val[1]

    def predict(self, X):
        Y = numpy.array([]).reshape(0, self.arch[-1])
        for x in X:
            y = numpy.array([self.predict_single_data(x)])
            Y = numpy.vstack((Y,y))
        return Y
numpy.random.seed(0)

nn = NeuralNetwork([2,2,1])
# Set the input data
X = numpy.array([[0, 0], [0, 1],
[1, 0], [1, 1]])
# Set the labels, the correct results for the xor operation
y = numpy.array([0, 1,
1, 0])

```

```
# Call the fit function and train the network for a chosen number of epochs
nn.fit(X, y, epochs=100000)
# Show the prediction results
print("Final prediction")
for s in X:
    print(s, nn.predict_single_data(s))
```

## **OUTPUT:**

```
epochs: 10000
epochs: 20000
epochs: 30000
epochs: 40000
epochs: 50000
epochs: 60000
epochs: 70000
epochs: 80000
epochs: 90000
epochs: 100000
Final prediction
[0 0] 2.769390318381638e-05
[0 1] 0.995154295769496
[1 0] 0.9951532983339081
[1 1] 2.7834280538588892e-05
```

## PRACTICAL NO. : 5

**AIM:** Write a program for linear separation.

**THEORY:** You could imagine that you have two attributes describing an edible object like a fruit for example: "sweetness" and "sourness". We could describe this by points in a two-dimensional space. The x axis for the sweetness and the y axis for the sourness. Imagine now that we have two fruits as points in this space, i.e. an orange at position (3.5, 1.8) and a lemon at (1.1, 3.9). We could define dividing lines to define the points which are more lemon-like and which are more orange-like. The following program calculates and renders a bunch of lines. The red ones are completely unusable for this purpose, because they are not separating the classes. Yet, it is obvious that even the green ones are not all useful.

### **CODE:**

```
import numpy as np

import matplotlib.pyplot as plt

def create_distance_function(a, b, c):
    def distance(x, y):
        """ returns tuple (d, pos)
        d is the distance
        If pos == -1 point is below the line,
        0 on the line and +1 if above the line
        """
        nom = a * x + b * y + c
        if nom == 0:
            pos = 0
        elif (nom < 0 and b < 0) or (nom > 0 and b > 0):
            pos = -1
```

```

    else:
        pos = 1
        return (np.absolute(nom) / np.sqrt( a ** 2 + b ** 2), pos)
    return distance

points = [ (3.5, 1.8), (1.1, 3.9) ]

fig, ax = plt.subplots()
ax.set_xlabel("sweetness")
ax.set_ylabel("sourness")
ax.set_xlim([-1, 6])
ax.set_ylim([-1, 8])
X = np.arange(-0.5, 5, 0.1)
colors = ["r", ""] # for the samples
size = 10

for (index, (x, y)) in enumerate(points):
    if index== 0:
        ax.plot(x, y, "o",
                color="darkorange",
                markersize=size)
    else:
        ax.plot(x, y, "oy",
                markersize=size)

step = 0.05
for x in np.arange(0, 1+step, step):
    slope = np.tan(np.arccos(x))

```

```

dist4line1 = create_distance_function(slope, -1, 0)
#print("x: ", x, "slope: ", slope)

Y = slope * X

results = []

for point in points:
    results.append(dist4line1(*point))

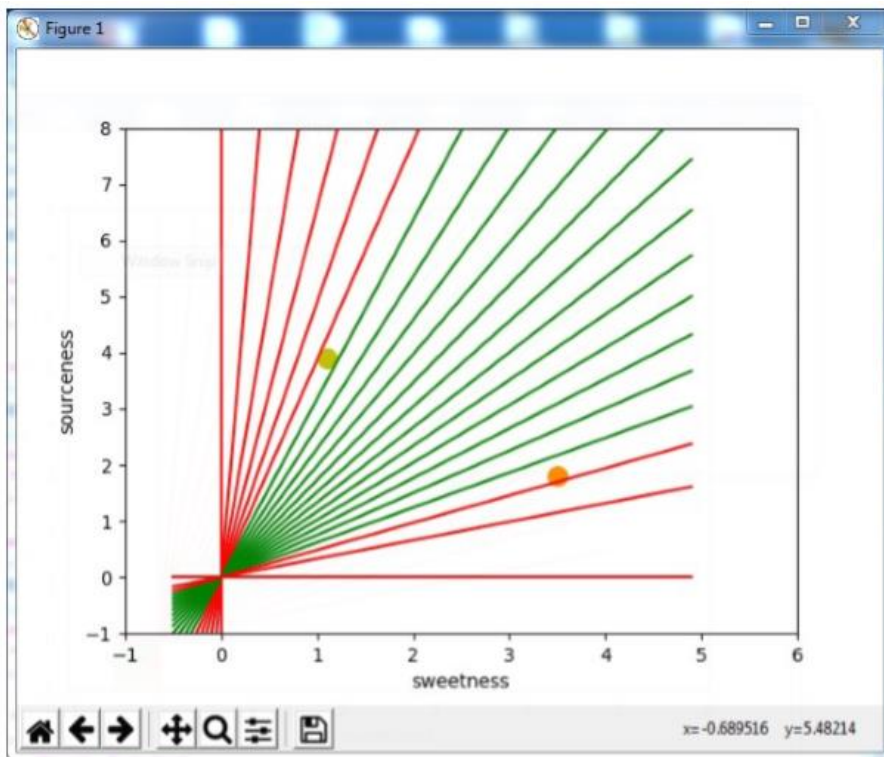
#print(slope, results)

if (results[0][1] != results[1][1]):
    ax.plot(X, Y, "g-")
else:
    ax.plot(X, Y, "r-")

plt.show()

```

## **OUTPUT:**





## **PRACTICAL NO. : 6**

**AIM:** Write a program to implement membership and identity operators is, is not.

### **CODE:**

```
#This is for integer data type
```

```
x = 5
```

```
if (type(x) is int):
```

```
    print ("true")
```

```
else:
```

```
    print ("false")
```

```
#This is for float data type
```

```
x = 5.2
```

```
if (type(x) is not int):
```

```
    print ("true")
```

```
else:
```

```
    print ("false")
```

### **OUTPUT:**

```
#For integer
```

```
x = 5
if (type(x) is int):
    print ("true")
else:
    print ("false")
|
```

```
=====
true
>>>
>>> |
```

#For float

```
x = 5.2
if (type(x) is not int):
    print ("true")
else:
    print ("false")
|
```

```
=====
true
>>>
```

## **PRACTICAL NO. : 7**

**AIM:** Write a program to implement membership and identity operators in, not in.

### **CODE:**

```
def overlapping(list1,list2):
```

```
    c=0
```

```
    d=0
```

```
    for i in list1:
```

```
        c+=1
```

```
    for i in list2:
```

```
        d+=1
```

```
    for i in range(0,c):
```

```
        for j in range(0,d):
```

```
            if(list1[i]==list2[j]):
```

```
                return 1
```

```
    return 0
```

```
list1=[1,2,3,4,5]
```

```
list2=[6,7,8,9]
```

```
if(overlapping(list1,list2)):
```

```
    print("overlapping")
```

```
else:
```

```
    print("not overlapping")
```

```
# Python program to illustrate
```

```

# Finding common member in list
# without using 'in' operator
# Define a function() that takes two lists
def overlapping(list1,list2):
    c=0
    d=0
    for i in list1:
        c+=1
    for i in list2:
        d+=1
    for i in range(0,c):
        for j in range(0,d):
            if(list1[i]==list2[j]):
                return 1
    return 0
list1=[1,2,3,4,5]
list2=[5,6,7,8,9]
if(overlapping(list1,list2)):
    print("overlapping")
else:
    print("not overlapping")

```

### **OUTPUT:**

```

===== RESTART: C:\Users\MSC_IT\Desktop\SC_Run_Programs\practical 9.py =====
not overlapping
overlapping

```

---