SANTOSH PARSE

APPLICATION ID: 113366

CLOUD COMPUTING

PRACTICAL JOURNAL

SUBJECT CODE: PSIT1P3

M.Sc. (IT)

PART-1 / SEM-1

“VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY, WADALA”

**AFFILIATED
TO
UNIVERSITY OF MUMBAI**

INSTITUTE OF DISTANCE AND OPEN LEARNING (IDOL)

CERTIFICATE

This is to certify that, **Santosh Parse** of M.Sc.(IT) Semester - I with Application ID **1 1 3 3 6 6** has completed the practical of **‘CLOUD COMPUTING’** in this college during the academic year **2022 - 2023**

Subject In-Charge **Coordinator -In-Charge**

Prof. Umesh Koyande

Examined By:

Table of Contents

Practical No: 1

A) Aim: Write a program for implementing client server communication model using TCP

Code:

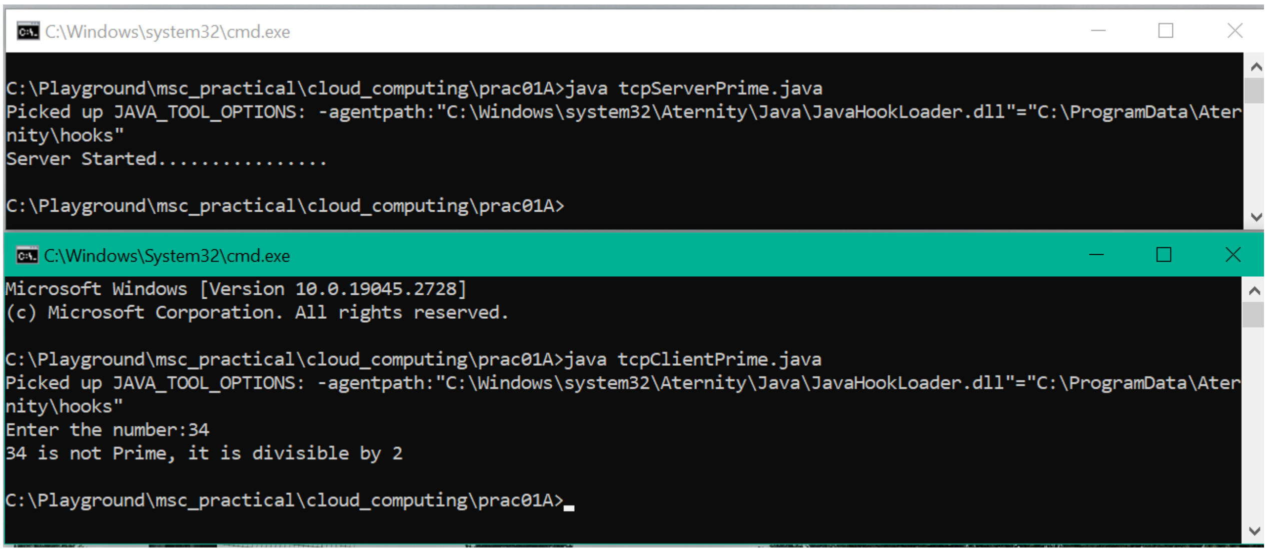
tcpClientPrime.java

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

public class tcpClientPrime {
   public static void main(String[] args) {
      try {
         Socket cs = new Socket("127.0.0.1", 8001);
         System.out.print("Enter the number:");
         BufferedReader infu = new BufferedReader(new InputStreamReader(System.in));

         int a = Integer.parseInt(infu.readLine());
         DataOutputStream out = new DataOutputStream(cs.getOutputStream());
         out.writeInt(a);

         DataInputStream in = new DataInputStream(cs.getInputStream());
         System.out.println(in.readUTF());
         cs.close();

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

tcpServerPrime.java

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

public class tcpServerPrime {
   public static void main(String[] args) {
      try {
         ServerSocket ss = new ServerSocket(8001);
         System.out.println("Server Started................");
         Socket s = ss.accept();

         DataInputStream in = new DataInputStream(s.getInputStream());
         int x = in.readInt();

         DataOutputStream out = new DataOutputStream(s.getOutputStream());
         int y = x / 2;

         if (x == 1 || x == 2 || x == 3) {
            out.writeUTF(x + " is Prime");
            System.exit(0);
         }

         boolean isPrime = true;

         for (int i = 2; i <= y; i++) {
            if (x % i == 0) {
                out.writeUTF(x + " is not Prime, it is divisible by " + i);
                isPrime = false;
                break;
            }
         }

         if (isPrime) {
            out.writeUTF(x + " is Prime");
         }

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

Output:



```
C:\Playground\msc_practical\cloud_computing\prac01A>java tcpServerPrime.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
Server Started................

C:\Playground\msc_practical\cloud_computing\prac01A>
```

```
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Playground\msc_practical\cloud_computing\prac01A>java tcpClientPrime.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
Enter the number:34
34 is not Prime, it is divisible by 2

C:\Playground\msc_practical\cloud_computing\prac01A>
```

B) Aim: A client server TCP based chatting application.
Code:

ChatClient.java

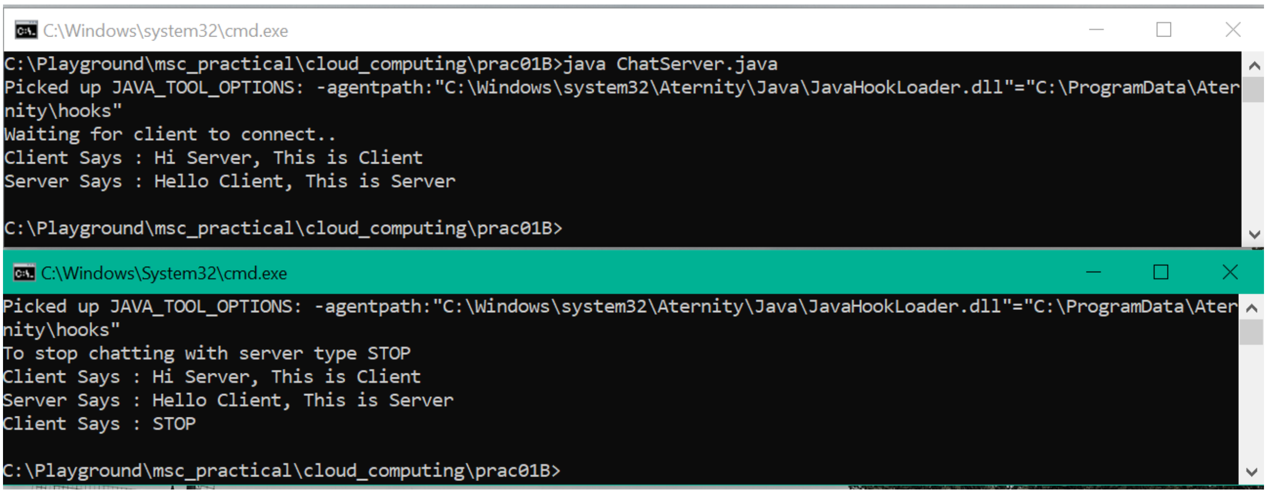
```
import java.net.*;
import java.io.*;

public class ChatClient {
   public static void main(String[] args) {
      try {
         Socket s = new Socket("Localhost", 8000);
         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
         DataOutputStream out = new DataOutputStream(s.getOutputStream());
         DataInputStream in = new DataInputStream(s.getInputStream());
         BufferedReader br2 = new BufferedReader(new InputStreamReader(in));
         String msg;
         System.out.println("To stop chatting with server type STOP");
         System.out.print("Client Says : ");
         while ((msg = br.readLine()) != null) {
            out.writeBytes(msg + "\n");
            if (msg.equals("STOP")) {
               break;
            }
            System.out.println("Server Says : " + br2.readLine());
            System.out.print("Client Says : ");
         }

         br.close();
         br2.close();
         in.close();
         out.close();
         s.close();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

ChatServer.java

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

public class ChatServer {
   public static void main(String[] args) {
      try {
         ServerSocket ss = new ServerSocket(8000);
         System.out.println("Waiting for client to connect..");
         Socket s = ss.accept();
```

```
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            DataOutputStream out = new DataOutputStream(s.getOutputStream());
            DataInputStream in = new DataInputStream(s.getInputStream());
            BufferedReader br2 = new BufferedReader(new InputStreamReader(in));
            String receive, send;
            while ((receive = br2.readLine()) != null) {
                if (receive.equals("STOP")) {
                    break;
                }
                System.out.println("Client Says : " + receive);
                System.out.print("Server Says : ");
                send = br.readLine();
                out.writeBytes(send + "\n");
            }

            br.close();
            br2.close();
            in.close();
            out.close();
            s.close();
            ss.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output:



```
C:\Playground\msc_practical\cloud_computing\prac01B>java ChatServer.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
Waiting for client to connect..
Client Says : Hi Server, This is Client
Server Says : Hello Client, This is Server

C:\Playground\msc_practical\cloud_computing\prac01B>
```

```
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
To stop chatting with server type STOP
Client Says : Hi Server, This is Client
Server Says : Hello Client, This is Server
Client Says : STOP

C:\Playground\msc_practical\cloud_computing\prac01B>
```

Practical No: 2

A) Aim: A client server-based program using UDP to find if the number entered is even or odd.

Code:

udpServerEO.java

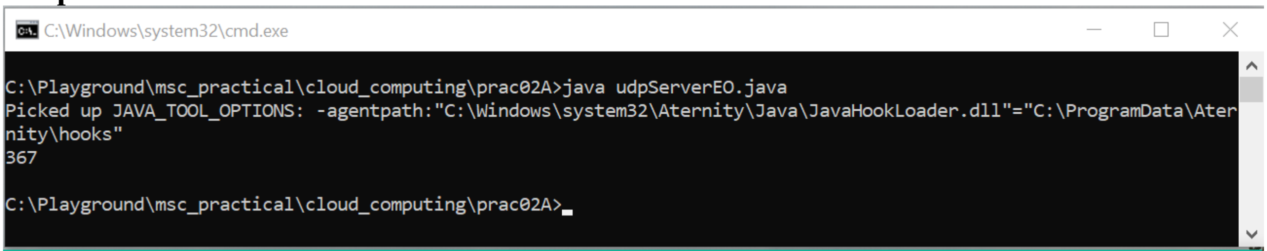
```
import java.net.*;

public class udpServerEO {
   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);

         ds.close();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

udpClientEO.java

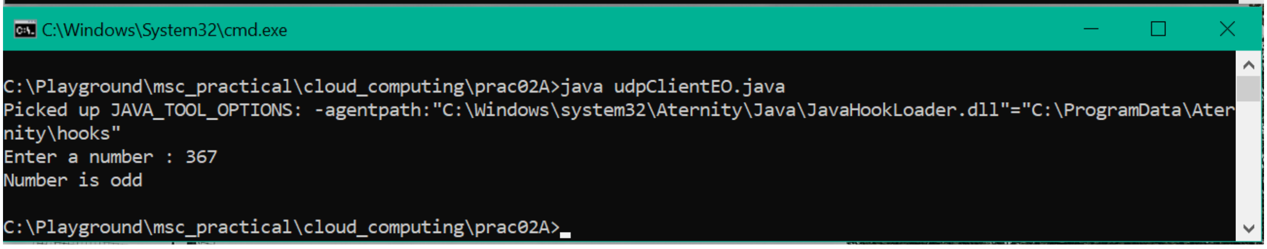
```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.*;

public class udpClientEO {
   public static void main(String[] args) {
      try {
         DatagramSocket ds = new DatagramSocket(1000);
         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
         System.out.print("Enter a number : ");
         String num = br.readLine();
```

```
            byte b[] = new byte[1024];
            b = num.getBytes();
            DatagramPacket dp = 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);

            ds.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output:



```
C:\Playground\msc_practical\cloud_computing\prac02A>java udpServerEO.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
367

C:\Playground\msc_practical\cloud_computing\prac02A>
```



```
C:\Playground\msc_practical\cloud_computing\prac02A>java udpClientEO.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
Enter a number : 367
Number is odd

C:\Playground\msc_practical\cloud_computing\prac02A>
```

B) Aim: A client server-based program using UDP to find the factorial of the entered number.
Code:

udpServerFact.java

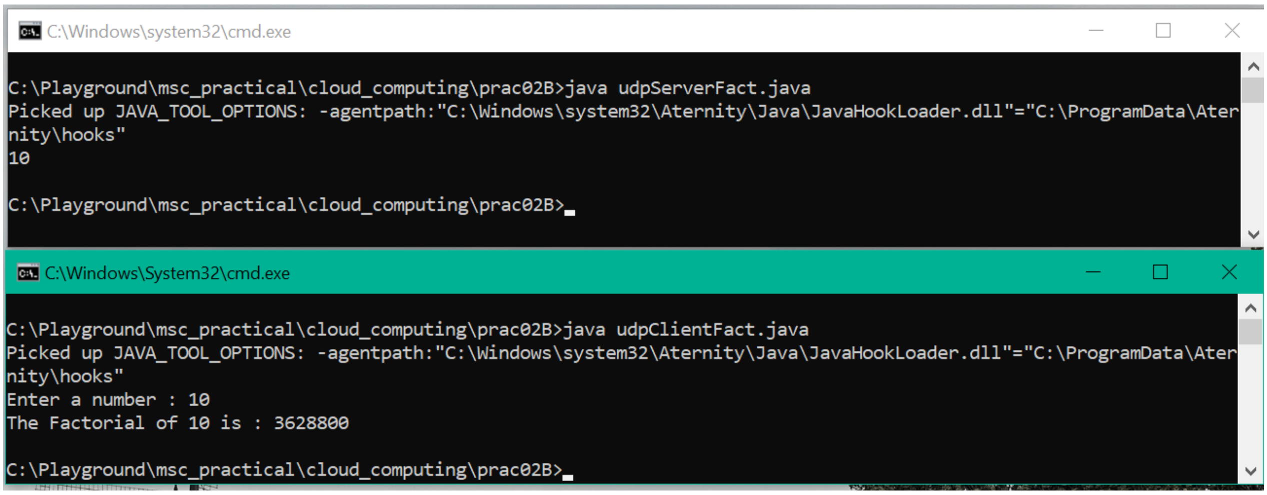
```
import java.net.*;

public class udpServerFact {

   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);
         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 " + str + " is : " + s;
         byte b1[] = new byte[1024];
         b1 = str1.getBytes();
         DatagramPacket dp1 = new DatagramPacket(b1, b1.length,
InetAddress.getLocalHost(), 1000);
         ds.send(dp1);

         ds.close();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

udpClientFact.java

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

public class udpClientFact {
   public static void main(String[] args) {
      try {
         DatagramSocket ds = new DatagramSocket(1000);
         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
         System.out.print("Enter a number : ");
         String num = br.readLine();
```

```
            byte b[] = new byte[1024];
            b = num.getBytes();
            DatagramPacket dp = 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);

            ds.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output:



```
C:\Playground\msc_practical\cloud_computing\prac02B>java udpServerFact.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
10

C:\Playground\msc_practical\cloud_computing\prac02B>
```

```
C:\Playground\msc_practical\cloud_computing\prac02B>java udpClientFact.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
Enter a number : 10
The Factorial of 10 is : 3628800

C:\Playground\msc_practical\cloud_computing\prac02B>
```

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

Code:

RPCServer.java

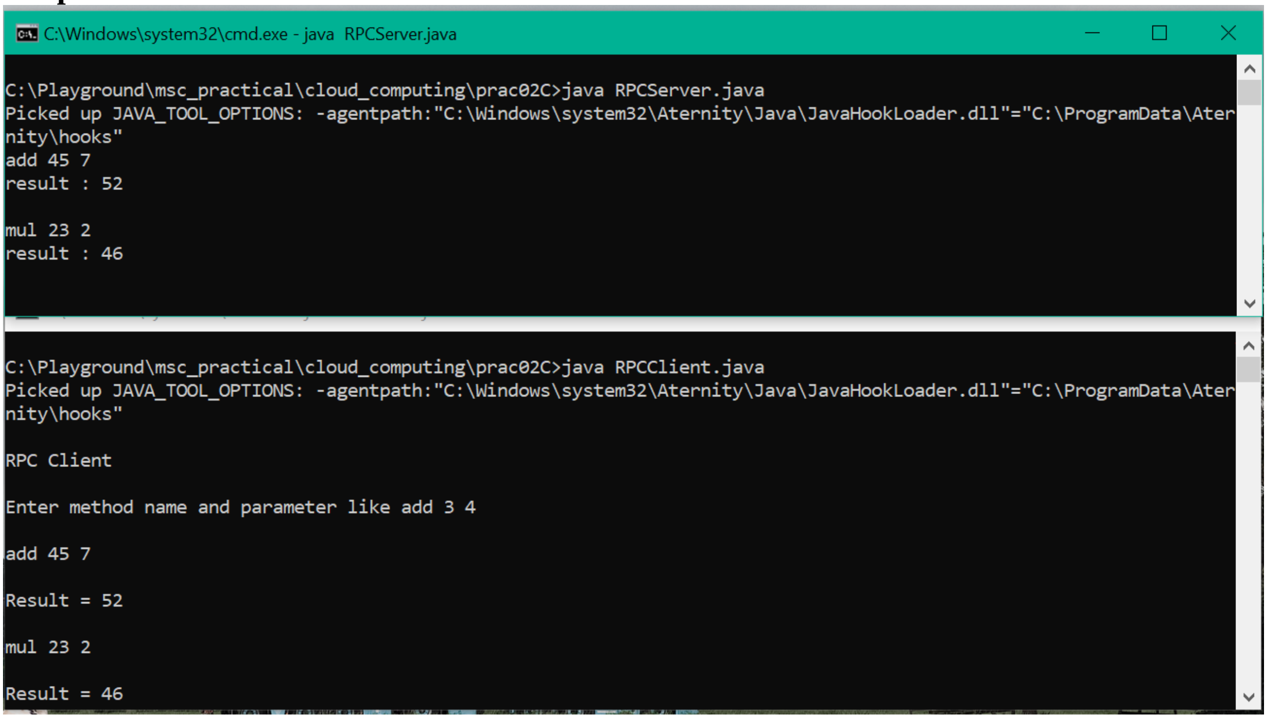
```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.StringTokenizer;

public class RPCServer {
    DatagramSocket ds;
    DatagramPacket dp;
    String str, methodName, result;
    int val1, val2;

    RPCServer() {
        try {
            ds = new DatagramSocket(1200);
            byte b[] = new byte[4096];
            while (true) {
                dp = new DatagramPacket(b, b.length);
                ds.receive(dp);
                str = new String(dp.getData(), 0, dp.getLength());
                if (str.equalsIgnoreCase("q")) {
                    System.exit(1);
                } else {
                    StringTokenizer st = new StringTokenizer(str, " ");
                    while (st.hasMoreTokens()) {
                        String token = st.nextToken();
                        methodName = token;
                        val1 = Integer.parseInt(st.nextToken());
                        val2 = Integer.parseInt(st.nextToken());
                    }
                }
                System.out.println(str);
                InetAddress ia = InetAddress.getLocalHost();
                if (methodName.equalsIgnoreCase("add")) {
                    result = "" + add(val1, val2);
                } else if (methodName.equalsIgnoreCase("sub")) {
                    result = "" + sub(val1, val2);
                } else if (methodName.equalsIgnoreCase("mul")) {
                    result = "" + mul(val1, val2);
                } else if (methodName.equalsIgnoreCase("div")) {
                    result = "" + div(val1, val2);
                }

                byte b1[] = result.getBytes();
                DatagramSocket ds1 = new DatagramSocket();
                DatagramPacket dp1 = new DatagramPacket(b1, b1.length,
InetAddress.getLocalHost(), 1300);
```

```
                System.out.println("result : " + result + "\n");
                ds1.send(dp1);
            }

        } catch (Exception e) {
            // TODO: handle exception
        }
    }

    public int add(int val1, int val2) {
        return val1 + val2;
    }

    public int sub(int val1, int val2) {
        return val1 - val2;
    }

    public int mul(int val1, int val2) {
        return val1 * val2;
    }

    public int div(int val1, int val2) {
        return val1 / val2;
    }

    public static void main(String[] args) {
        new RPCServer();
    }
}
```

RPCClient.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class RPCClient {
    RPCClient() {
        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 like 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();
        }
    }

    public static void main(String[] args) {
        new RPCClient();
    }
}
```

Output:



```
C:\Playground\msc_practical\cloud_computing\prac02C>java RPCServer.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
add 45 7
result : 52

mul 23 2
result : 46
```

```
C:\Playground\msc_practical\cloud_computing\prac02C>java RPCClient.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"

RPC Client

Enter method name and parameter like add 3 4

add 45 7

Result = 52

mul 23 2

Result = 46
```

D) Aim: A program that finds the square, square root, cube and cube root of the entered number.

Code:

RPCNumServer.java

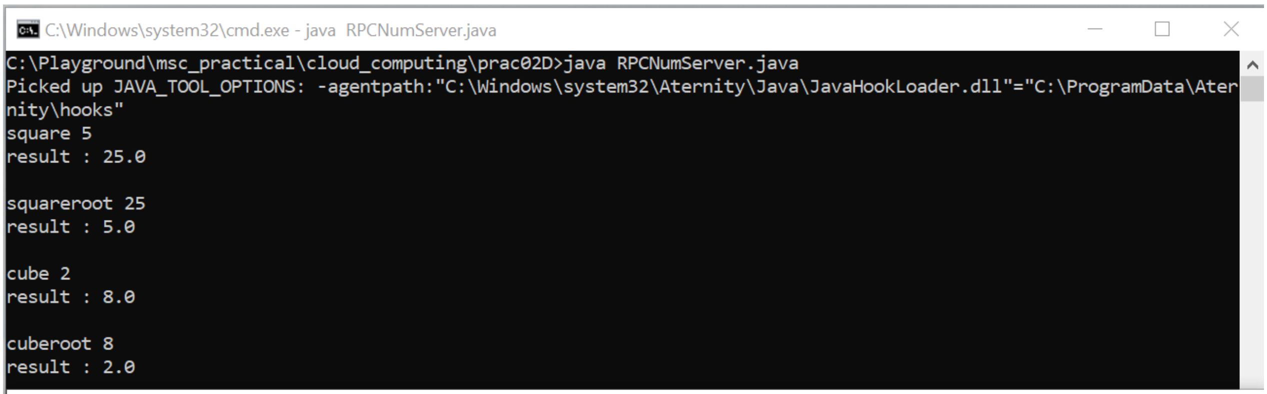
```
import java.util.*;
import java.net.*;

public class RPCNumServer {
    DatagramSocket ds;
    DatagramPacket dp;
    String str, methodName, result;
    int val;

    RPCNumServer() {
        try {
            ds = new DatagramSocket(1200);
            byte b[] = new byte[4096];
            while (true) {
                dp = new DatagramPacket(b, b.length);
                ds.receive(dp);
                str = new String(dp.getData(), 0, dp.getLength());
                if (str.equalsIgnoreCase("q")) {
                    System.exit(1);
                } else {
                    StringTokenizer st = new StringTokenizer(str, " ");
                    int i = 0;
                    while (st.hasMoreTokens()) {
                        String token = st.nextToken();
                        methodName = token;
                        val = Integer.parseInt(st.nextToken());
                    }
                }
                System.out.println(str);
                InetAddress ia = InetAddress.getLocalHost();
                if (methodName.equalsIgnoreCase("square")) {
                    result = "" + square(val);
                } else if (methodName.equalsIgnoreCase("squareroot")) {
                    result = "" + squareroot(val);
                } else if (methodName.equalsIgnoreCase("cube")) {
                    result = "" + cube(val);
                } else if (methodName.equalsIgnoreCase("cuberoot")) {
                    result = "" + cuberoot(val);
                }
                byte b1[] = result.getBytes();
                DatagramSocket ds1 = new DatagramSocket();
                DatagramPacket dp1 = new DatagramPacket(b1, b1.length,
InetAddress.getLocalHost(), 1300);
                System.out.println("result : " + result + "\n");
                ds1.send(dp1);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public double square(int a) throws Exception {
        double ans;
        ans = a * a;
        return ans;
    }

    public double squareroot(int a) throws Exception {
        double ans;
        ans = Math.sqrt(a);
        return ans;
    }

    public double cube(int a) throws Exception {
        double ans;
        ans = a * a * a;
        return ans;
    }

    public double cuberoot(int a) throws Exception {
        double ans;
        ans = Math.cbrt(a);
        return ans;
    }

    public static void main(String[] args) {
        new RPCNumServer();
    }
}
```

RPCNumClient.java

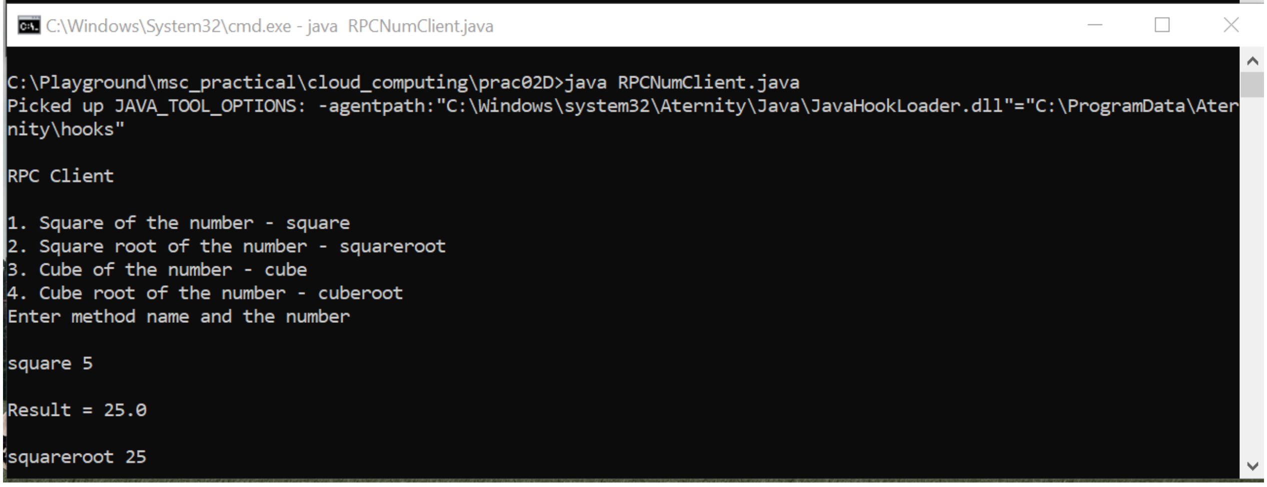
```
import java.io.*;
import java.net.*;

public class RPCNumClient {
    RPCNumClient() {
        try {
            InetAddress ia = InetAddress.getLocalHost();
            DatagramSocket ds = new DatagramSocket();
            DatagramSocket ds1 = new DatagramSocket(1300);
            System.out.println("\nRPC Client\n");
            System.out.println(
                    "1. Square of the number - square\n2. Square root of the number - squareroot\n3.
Cube of the number - cube\n4. Cube root of the number - cuberoot");
```

```
            System.out.println("Enter method name and the number\n");
            while (true) {
                BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
                String str = br.readLine();
                byte b[] = str.getBytes();
                DatagramPacket dp = new DatagramPacket(b, b.length, ia, 1200);
                ds.send(dp);
                dp = new DatagramPacket(b, b.length);
                ds1.receive(dp);
                String s = new String(dp.getData(), 0, dp.getLength());
                System.out.println("\nResult = " + s + "\n");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new RPCNumClient();
    }
}
```

Output:



```
C:\Playground\msc_practical\cloud_computing\prac02D>java RPCNumServer.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
square 5
result : 25.0

squareroot 25
result : 5.0

cube 2
result : 8.0

cuberoot 8
result : 2.0
```



```

C:\Playground\msc_practical\cloud_computing\prac02D>java RPCNumClient.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"

RPC Client

1. Square of the number - square
2. Square root of the number - squareroot
3. Cube of the number - cube
4. Cube root of the number - cuberoot
Enter method name and the number

square 5

Result = 25.0

squareroot 25
```

Practical No:3

Aim: A multicast Socket example.
Code:

BroadcastServer.java

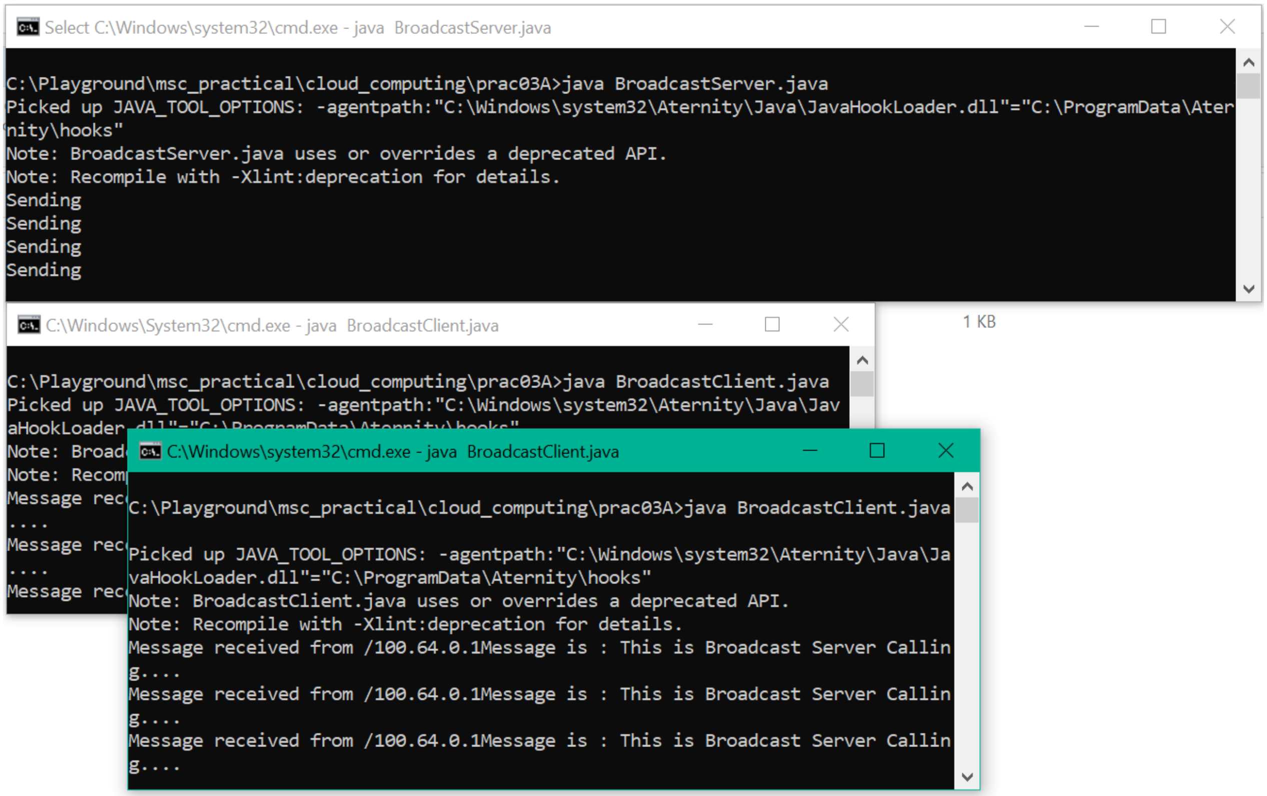
```
import java.net.*;
import java.io.*;
import java.util.*;

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 Pushpa Calling....");
         data = str.getBytes();
         packet = new DatagramPacket(data, str.length(), address, PORT);
         // Sends the packet
         socket.send(packet);
      }
   }
}
```

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 (;;) {
           socket.receive(packet);
           String str = new String(packet.getData());
           System.out.println("Message received from " + packet.getAddress() + "Message is : "
+ str);
        }
    }
}
```

Output:


Select C:\Windows\system32\cmd.exe - java BroadcastServer.java
C:\Windows\System32\cmd.exe - java BroadcastClient.java
1 KB
C:\Windows\system32\cmd.exe - java BroadcastClient.java

```
C:\Playground\msc_practical\cloud_computing\prac03A>java BroadcastServer.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Ater
nity\hooks"
Note: BroadcastServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Sending
Sending
Sending
Sending
```

```
C:\Playground\msc_practical\cloud_computing\prac03A>java BroadcastClient.java
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\Jav
aHookLoader.dll"="C:\ProgramData\Aternity\hooks"
Note: Broad
Note: Recom
Message rec
....
Message rec
....
Message rec
```

```
C:\Playground\msc_practical\cloud_computing\prac03A>java BroadcastClient.java

Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\Ja
vaHookLoader.dll"="C:\ProgramData\Aternity\hooks"
Note: BroadcastClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Message received from /100.64.0.1Message is : This is Broadcast Server Callin
g....
Message received from /100.64.0.1Message is : This is Broadcast Server Callin
g....
Message received from /100.64.0.1Message is : This is Broadcast Server Callin
g....
```

Practical No: 4

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

Code:

InterDate.java

```
import java.rmi.*;

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

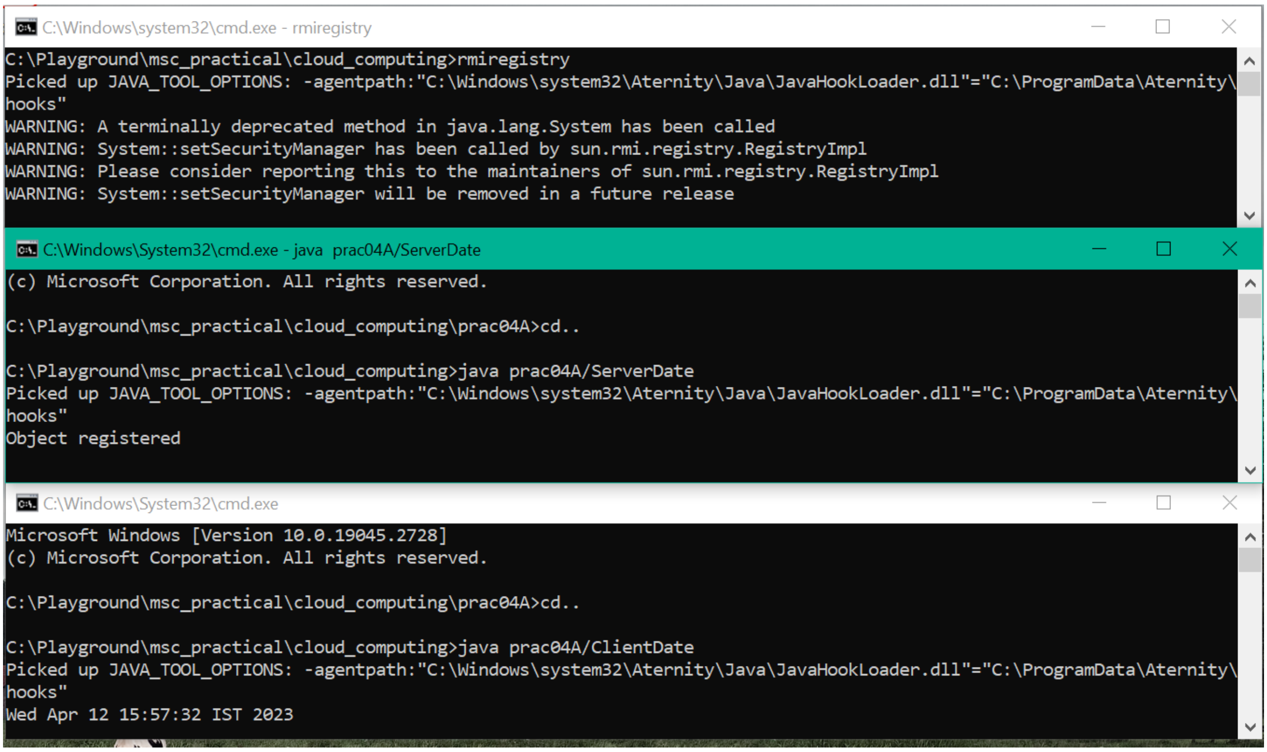
ServerDate.java

```
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("RMI2", s1);
      System.out.println("Object registered");
   }

}
```

ClientDate.java

```
import java.rmi.*;

public class ClientDate {
   public static void main(String[] args) throws Exception {
      String s1;
      InterDate h1 = (InterDate) Naming.lookup("RMI2");
      s1 = h1.display();
      System.out.println(s1);
   }
}\
```

Output:



```
C:\Playground\msc_practical\cloud_computing>rmiregistry
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Aternity\
hooks"
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by sun.rmi.registry.RegistryImpl
WARNING: Please consider reporting this to the maintainers of sun.rmi.registry.RegistryImpl
WARNING: System::setSecurityManager will be removed in a future release
```

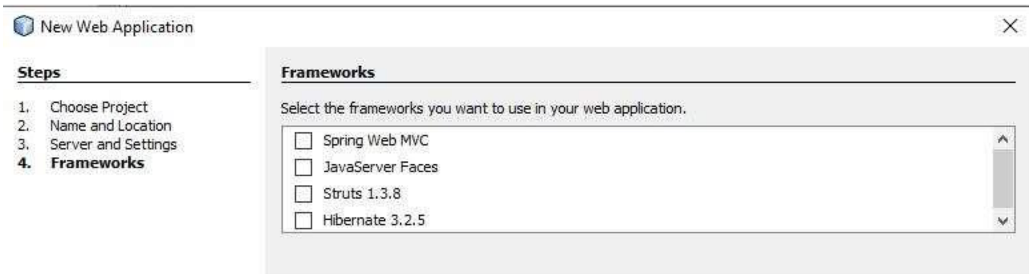
```
(c) Microsoft Corporation. All rights reserved.

C:\Playground\msc_practical\cloud_computing\prac04A>cd..

C:\Playground\msc_practical\cloud_computing>java prac04A/ServerDate
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Aternity\
hooks"
Object registered
```

```
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Playground\msc_practical\cloud_computing\prac04A>cd..

C:\Playground\msc_practical\cloud_computing>java prac04A/ClientDate
Picked up JAVA_TOOL_OPTIONS: -agentpath:"C:\Windows\system32\Aternity\Java\JavaHookLoader.dll"="C:\ProgramData\Aternity\
hooks"
Wed Apr 12 15:57:32 IST 2023
```

Practical No: 5

Aim: Show the implementation of web services

Step 1: - File-> New Project -> Choose Project -> Java Web -> Web Application -> Next-> Give Project Name -> Next -> Next -> Finish



New Web Application

Steps

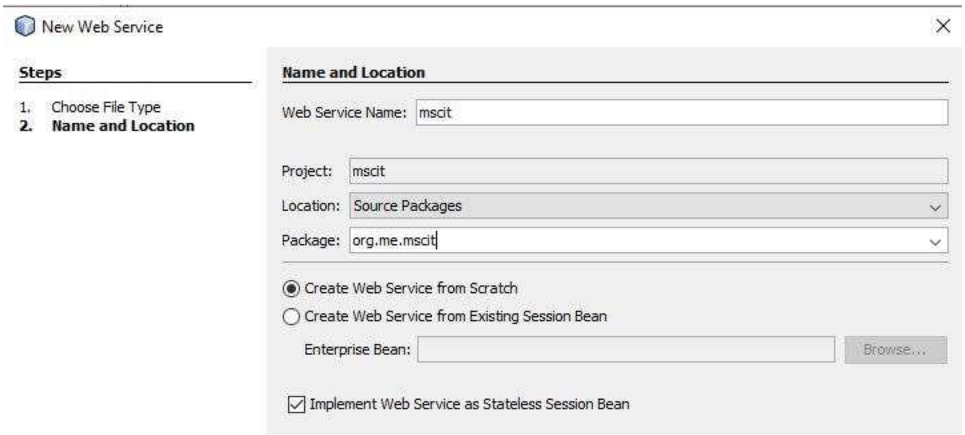
1. Choose Project
2. Name and Location
3. Server and Settings
4. **Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

□ Spring Web MVC
□ JavaServer Faces
□ Struts 1.3.8
□ Hibernate 3.2.5

Step 2: - Right Click on Project -> New -> Web Service -> Give name and package name -> Finish



New Web Service

Steps

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

Name and Location

Web Service Name: mscit

Project: mscit

Location: Source Packages

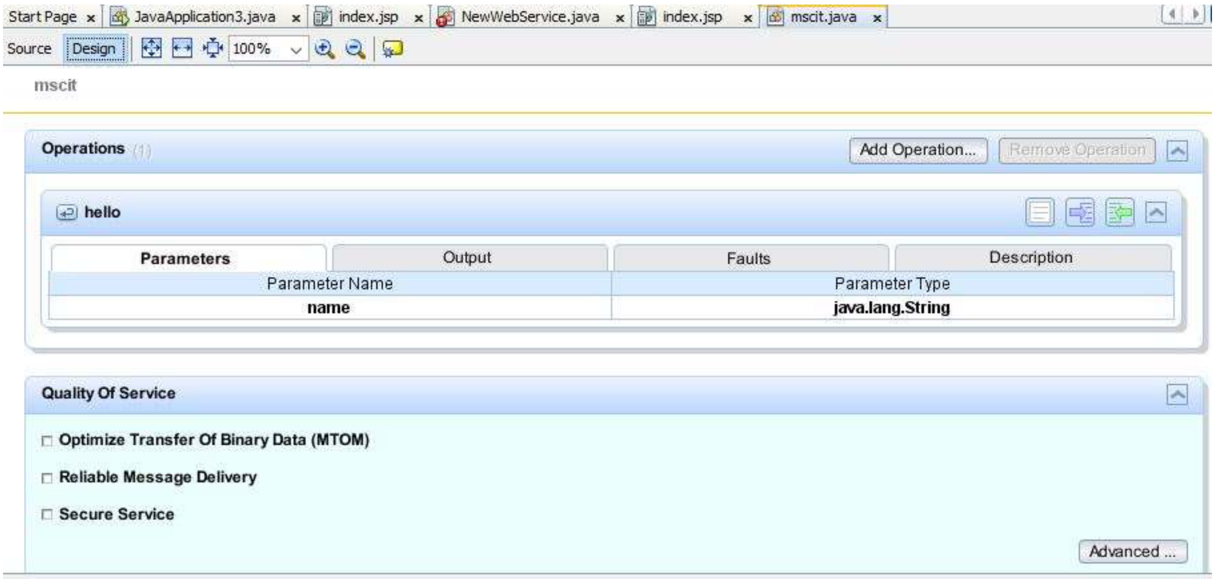
Package: org.me.mscit

● Create Web Service from Scratch
○ Create Web Service from Existing Session Bean

Enterprise Bean: Browse...

☑ Implement Web Service as Stateless Session Bean

Step 3: - Go on Design Section of .java file -> add operation as add and return type as int -> add two parameters i & j -> OK



mscit

Operations (1)

Add Operation... Remove Operation

hello

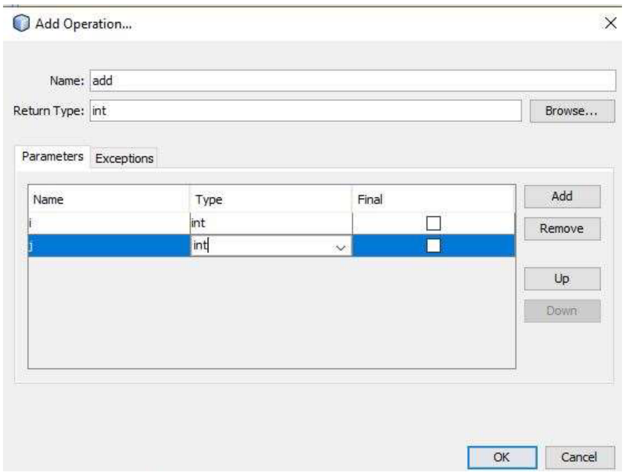
Parameters	Output	Faults	Description

Parameter Name	Parameter Type
name	java.lang.String

Quality Of Service

□ Optimize Transfer Of Binary Data (MTOM)
□ Reliable Message Delivery
□ Secure Service

Advanced ...



Add Operation...

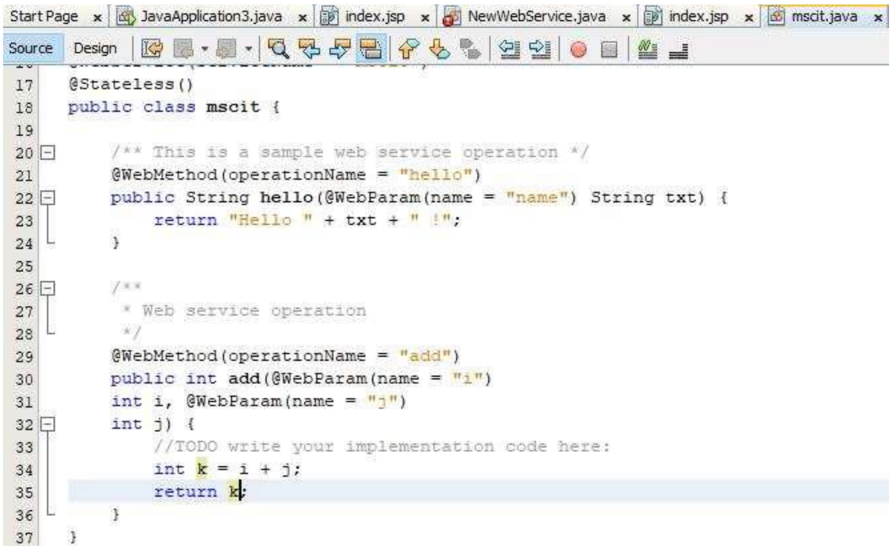
Name: add

Return Type: int

Parameters | Exceptions

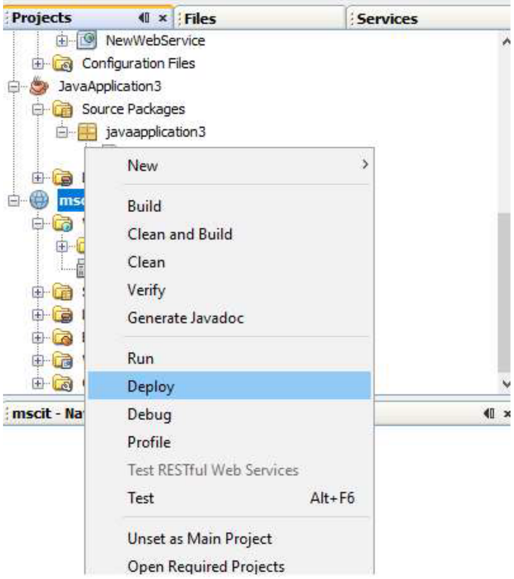
Name	Type	Final
i	int	□
j	int	□

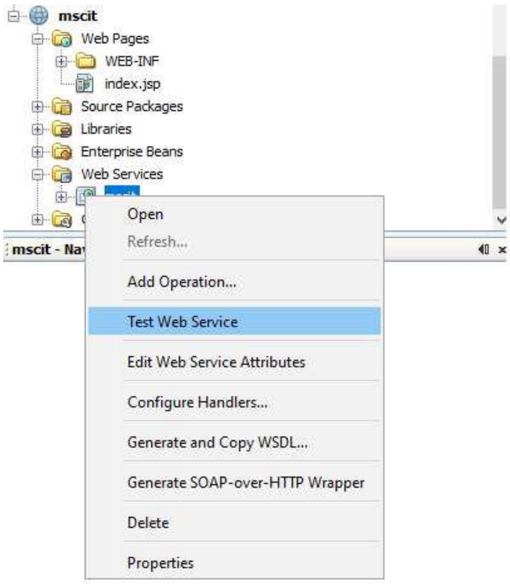
Step 4: - Add following code inside .java file



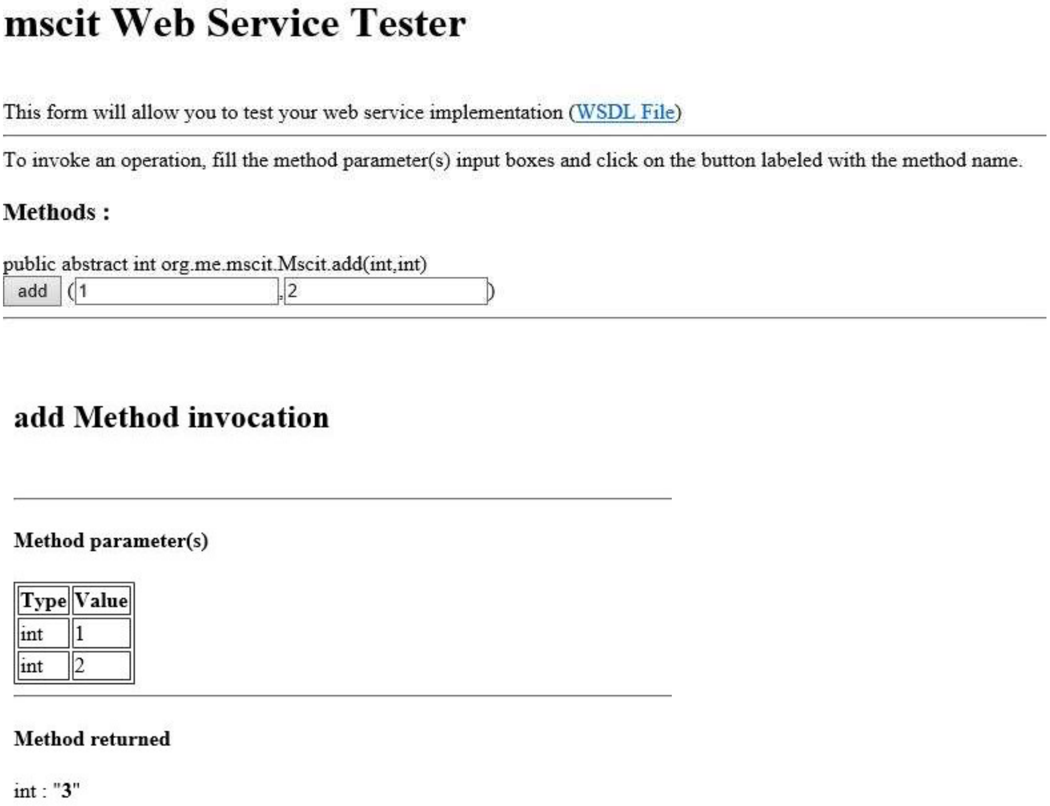
```
@Stateless()
public class mscit {

    /** This is a sample web service operation */
    @WebMethod(operationName = "hello")
    public String hello(@WebParam(name = "name") String txt) {
        return "Hello " + txt + " !";
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "add")
    public int add(@WebParam(name = "i")
    int i, @WebParam(name = "j")
    int j) {
        //TODO write your implementation code here:
        int k = i + j;
        return k;
    }
}
```

Step 5: - Right click on the project name and Deploy the Web Service and finally Test the Web Service





Output:



mscit 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.mscit.Mscit.add(int,int)

add (1 ,2)

add Method invocation

Method parameter(s)

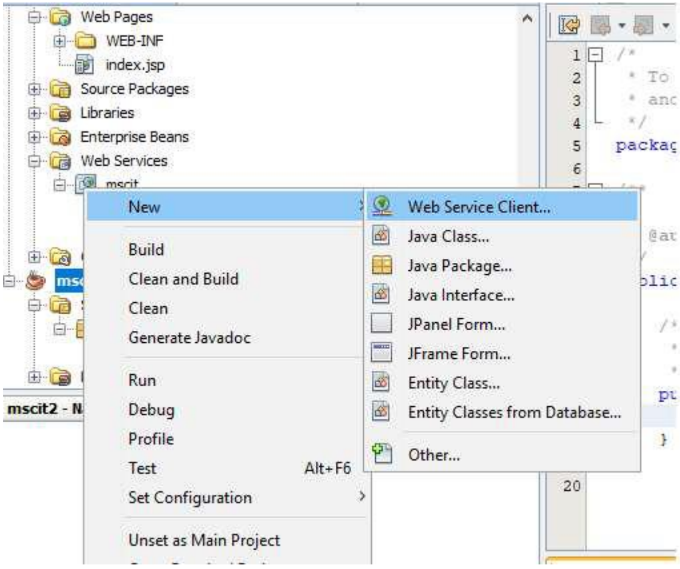
Type	Value
int	1
int	2

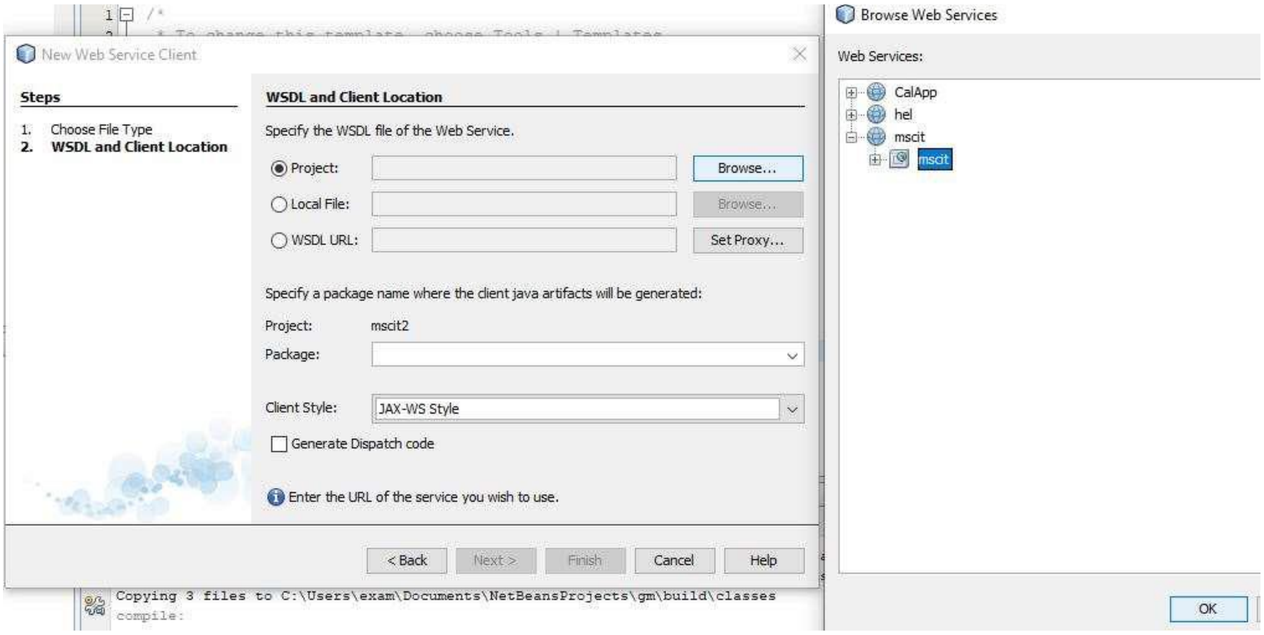
Method returned

int : "3"

Steps for Consuming the Web Service:

Step 1: - Create new java application Right click on project -> new -> Web Service Client -> browse Web Service -> select the deployed project -> OK





WSDL and Client Location

Specify the WSDL file of the Web Service.

● Project:

○ Local File:

○ WSDL URL:

Specify a package name where the client java artifacts will be generated:

Project: mscit2

Package:

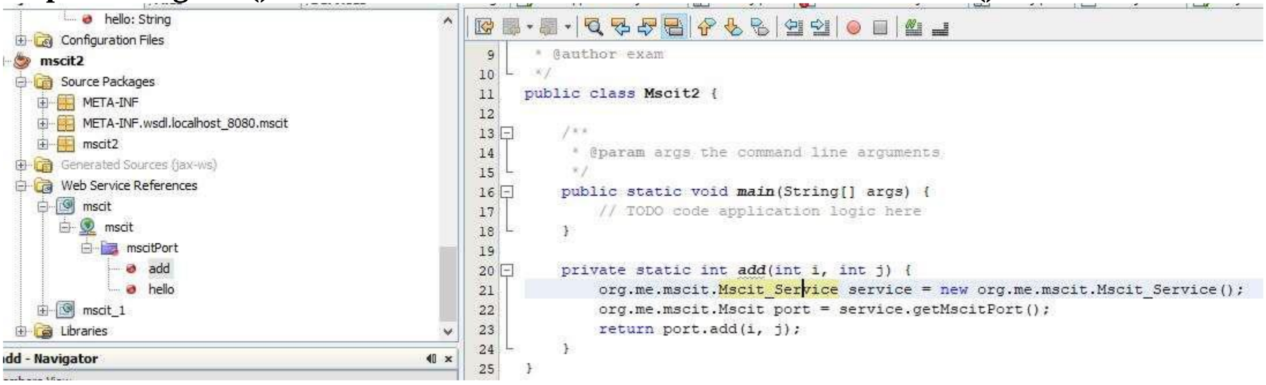
Client Style: JAX-WS Style

□ Generate Dispatch code

Enter the URL of the service you wish to use.

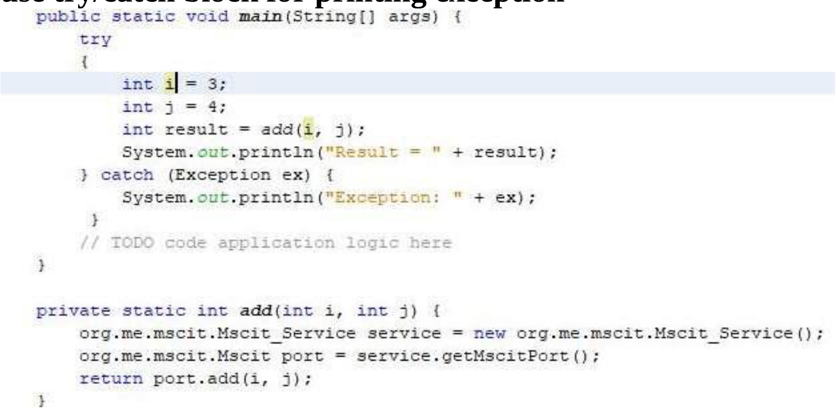
```
Copying 3 files to C:\Users\exam\Documents\NetBeansProjects\gm\build\classes
compile:
```

Step 2: - drag add() node from Web Service References below the Main() method



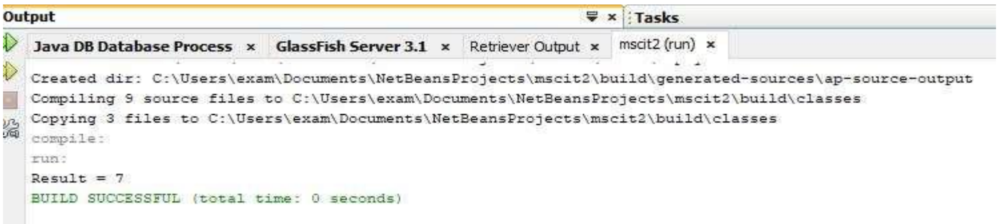
```
 * @author exam
 */
public class Mscit2 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

    private static int add(int i, int j) {
        org.me.mscit.Mscit_Service service = new org.me.mscit.Mscit_Service();
        org.me.mscit.Mscit port = service.getMscitPort();
        return port.add(i, j);
    }
}
```

use try/catch block for printing 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);
     }
    // TODO code application logic here
}

private static int add(int i, int j) {
    org.me.mscit.Mscit_Service service = new org.me.mscit.Mscit_Service();
    org.me.mscit.Mscit port = service.getMscitPort();
    return port.add(i, j);
}
```

Output: -



```
Created dir: C:\Users\exam\Documents\NetBeansProjects\mscit2\build\generated-sources\ap-source-output
Compiling 9 source files to C:\Users\exam\Documents\NetBeansProjects\mscit2\build\classes
Copying 3 files to C:\Users\exam\Documents\NetBeansProjects\mscit2\build\classes
compile:
run:
Result = 7
BUILD SUCCESSFUL (total time: 0 seconds)
```