PRACTICAL 2

Aim: Write a program for implementing Client Sever communication model using UDP.

Practical 2A: A client server based program using UDP to find if the number entered is even or odd.

Code:

Pramod_248637_udpServerEO.java

```
import java.io.*;
import java.net.*;
public class Pramod_248637_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("Pramod 248637: "+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();
```

```
}
}
}
```

Pramod_248637_udpClientEO.java

```
import java.io.*;
import java.net.*;
public class Pramod_248637_udpClientEO{
public static void main(String[] args){
try{
DatagramSocket ds = new DatagramSocket(1000);
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Pramod 248637: 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);
}catch(Exception e){
e.printStackTrace();
}
}
}
```

Output:

```
C:\Users\admin\Desktop\temp>java Pramod_248637_udpServerEO
Pramod 248637: 32352354
```

```
C:\Users\admin\Desktop\temp>java Pramod_248637_udpClientEO
Pramod 248637: Enter a number:
32352354
Number is even
```

Practical 2B: A client server based program using UDP to find the factorial of the entered number

Code:

Pramod_248637_udpServerFact.java

```
import java.io.*;
import java.net.*;
public class Pramod_248637_udpServerFact{
public static void main(String[] args){
try{
DatagramSocket ds = new DatagramSocket(2000);
byte b[] = new byte[4096];
DatagramPacket dp = new DatagramPacket(b, b.length);
ds.receive(dp);
String s = new String(dp.getData(),0,dp.getLength());
int num = Integer.parseInt(s);
int fac = 1;
while(num > 1){
fac = fac * num;
num = num - 1;
}

String str_fac =Integer.toString(fac);
byte b1[] = new byte[4096];
b1 = str_fac.getBytes();
DatagramPacket dp1 = new DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
```

```
ds.send(dp1);
}catch(Exception e){
e.printStackTrace();
}
}
}
```

Pramod_248637_udpClientFact.java

```
import java.io.*;
import java.net.*;
public class Pramod_248637_udpClientFact{
public static void main(String[] args){
try{
DatagramSocket ds = new DatagramSocket(1000);
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Pramod 248637: Enter a number: ");
String num = br.readLine();
byte b[] = new byte[4096];
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("Factorial: "+str);
}catch(Exception e){
e.printStackTrace();
}
}
```

}

Output:

```
C:\Users\admin\Desktop\temp>javac Pramod_248637_udpServerFact.java

C:\Users\admin\Desktop\temp>java Pramod_248637_udpServerFact
```

```
C:\Users\admin\Desktop\temp>java Pramod_248637_udpClientFact
Pramod 248637: Enter a number:
5
Factorial: 120
```

Practical 2C: A program to implement simple calculator operations like addition, subtraction, multiplication and division

Code:

Pramod_248637_RPCServer.java

```
import java.io.*;
import java.net.*;
import java.util.*;
public class Pramod_248637_RPCServer{
DatagramSocket ds;
DatagramPacket dp;
String str, methodName, result;
int val1, val2;

Pramod_248637_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{
System.out.println(str);
StringTokenizer st = new StringTokenizer(str," ");
int i = 0;
while (st.hasMoreTokens()){
String token = st.nextToken();
methodName = token;
val1 = Integer.parseInt(st.nextToken());
val2 = Integer.parseInt(st.nextToken());
}
}
System.out.println(str);
InetAddress ia = InetAddress.getLocalHost();
if(methodName.equalsIgnoreCase("add")){
int res = val1 + val2;
result = ""+res;
}else if(methodName.equalsIgnoreCase("sub")){
int res = val1 - val2;
result = ""+res;
}else if(methodName.equalsIgnoreCase("mul")){
int res = val1 * val2;
result = ""+res;
}else if(methodName.equalsIgnoreCase("div")){
int res = val1 / val2;
result = ""+res;
}
byte b1[] = result.getBytes();
DatagramSocket ds1 = new DatagramSocket();
DatagramPacket dp1 = new Dgth, InetAddress.getLocalHost(),1300);
```

```
System.out.println("Result: "+result);
ds1.send(dp1);
ds1.close();
}
}catch(Exception e){
e.printStackTrace();
}
}
public static void main(String[] args){
new Pramod_248637_RPCServer();
}
}
```

Pramod_248637_RPCClient.java

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

public class Pramod_248637_RPCClient{
Pramod_248637_RPCClient(){
try{
InetAddress ia = InetAddress.getLocalHost();
DatagramSocket ds = new DatagramSocket();
DatagramSocket ds1 = new DatagramSocket(1300);
System.out.println("Pramod_248637: RPC Client");
System.out.println("Enter method name and parameter like add 3 4");
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("Result = "+s+"\n");
}
}catch(Exception e){
e.printStackTrace();
}
}
public static void main(String[] args){
        new Pramod_248637_RPCClient();
}
}
```

Output:

```
C:\Users\admin\Desktop\temp>java Pramod_248637_RPCClient
Pramod_248637: RPC Client
Enter method name and parameter like add 3 4
add 3 4
Result = 7

mul 20 10
Result = 200
```

```
C:\Users\admin\Desktop\temp>java Pramod_248637_RPCServer
add 3 4
Result: 7
mul 20 10
Result: 200
```

Practical 2D: A program that finds the square, square root, cube and cube root of the entered number.

Pramod_248637_RPCNumServer.java

```
import java.util.*;
import java.net.*;
class Pramod_248637_RPCNumServer
```

```
{
   DatagramSocket ds;
   DatagramPacket dp;
   String str,methodName,result;
   int val;
   Pramod_248637_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 Pramod_248637_RPCNumServer();
}
}
```

Pramod_248637_RPCNumClient.java

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

class Pramod_248637_RPCNumClient {
   Pramod_248637_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 Pramod_248637_RPCNumClient();
    }
}
```

Output:

```
PS F:\College\Sem6\CC\Prac\Prac2> java Pramod_248637_RPCNumServer.java
Pramod Joshi 248637

square 2
result : 4.0

Pramod Joshi 248637

squareroot 4
result : 2.0

Pramod Joshi 248637

cube 32
result : 32768.0

Pramod Joshi 248637

cuberoot 8
result : 2.0
```

```
PS F:\College\Sem6\CC\Prac\Prac2> java Pramod_248637_RPCNumClient.java

RPC Client

Pramod Joshi 248637

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 2

Result = 4.0

squareroot 4

Result = 2.0

cube 32

Result = 32768.0

cuberoot 8

Result = 2.0
```