**Name – Yashchandra Upadhyay**

**Roll no. 05**

**Practical 1 Implement Client Server communication model using TCP**

**Question No. 1 check prime**

**Client Program:**

/\*Client program to check prime or not \*/

import java.net.\*;

import java.io.\*;

class tcpClientPrime

{

public static void main(String args[])

{

try

{

Socket cs = new Socket("LocalHost",8001); BufferedReader infu = new BufferedReader(new

InputStreamReader(System.in));

System.out.println("Enter a number : ");

int a = Integer.parseInt(infu.readLine());

DataOutputStream out = new

DataOutputStream(cs.getOutputStream());

out.writeInt(a);

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

}

catch(Exception e)

{

System.out.println(e.toString());

}

}

}

**Server Program:**

/\* Server program to check given no is prime or not in response to client request \*/

import java.net.\*;

import java.io.\*;

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

if(x ==1 || x ==2 || x ==3)

{

otc.writeUTF(x + "is Prime");

System.exit(0);

}

for(int i=2; i<=y; i++)

{

if(x%i != 0)

{

otc.writeUTF(x + " is Prime");

}

else

{

otc.writeUTF(x + " is not Prime");

}

}

}

catch(Exception e)

{

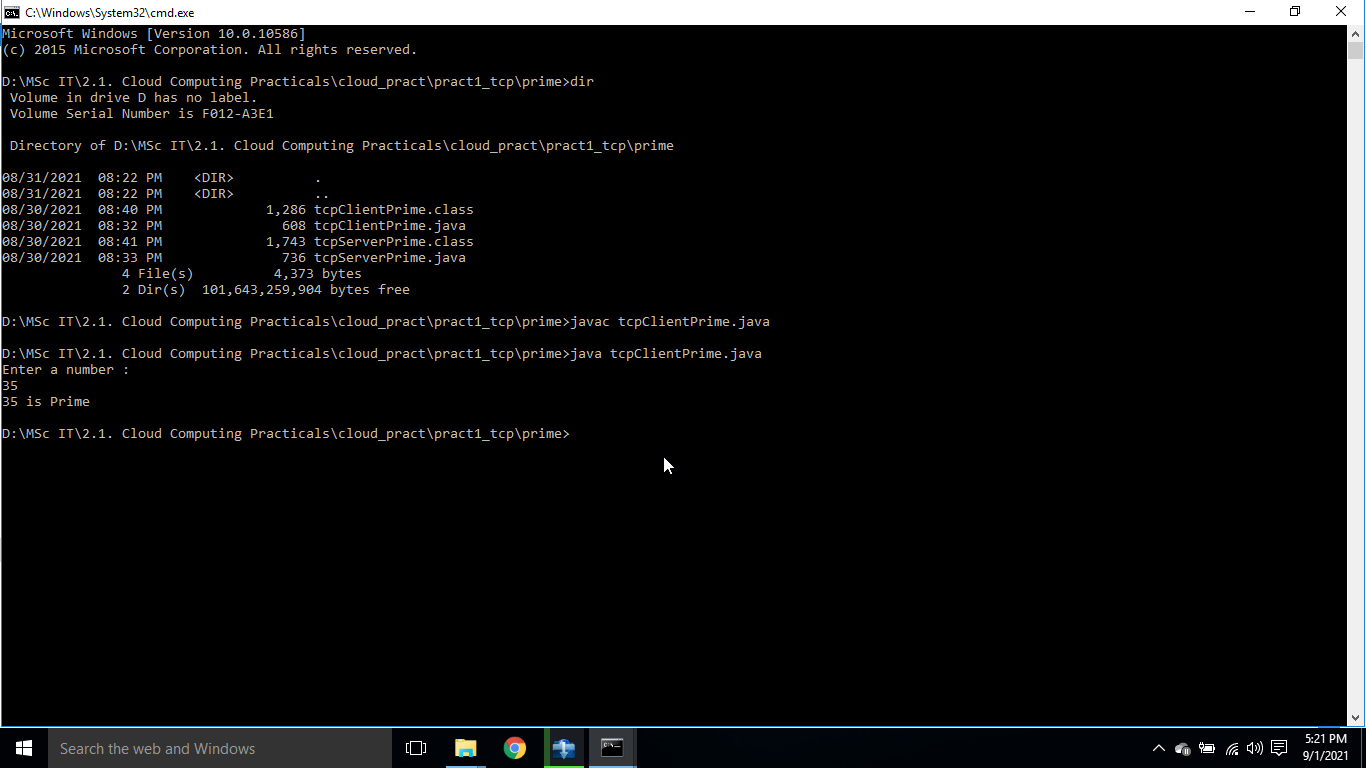
System.out.println(e.toString());

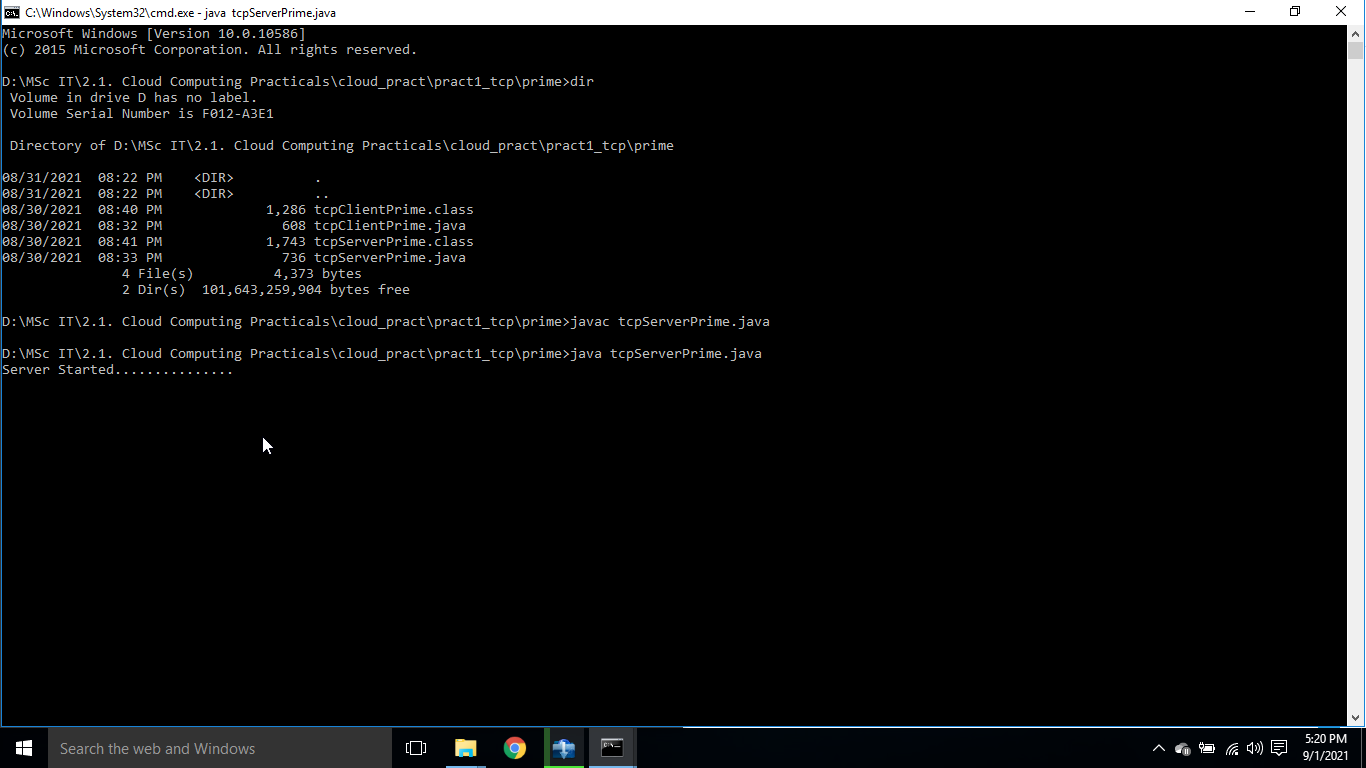
}

}

}

**Output:**





**Question No. 2 chat between server and client**

**Client Program:**

/\*Program to implement chat client \*/

import java.net.\*;

import java.io.\*;

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());

BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

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 : "+in.readLine());

System.out.print("Client Says : ");

}

br.close();

in.close();

out.close();

s.close();

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

**Server Program:**

/\* Program to implement  chat server \*/

import java.net.\*;

import java.io.\*;

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());

BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

String receive, send;

while((receive = in.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();

in.close();

out.close();

s.close();

}

catch(Exception e)

{

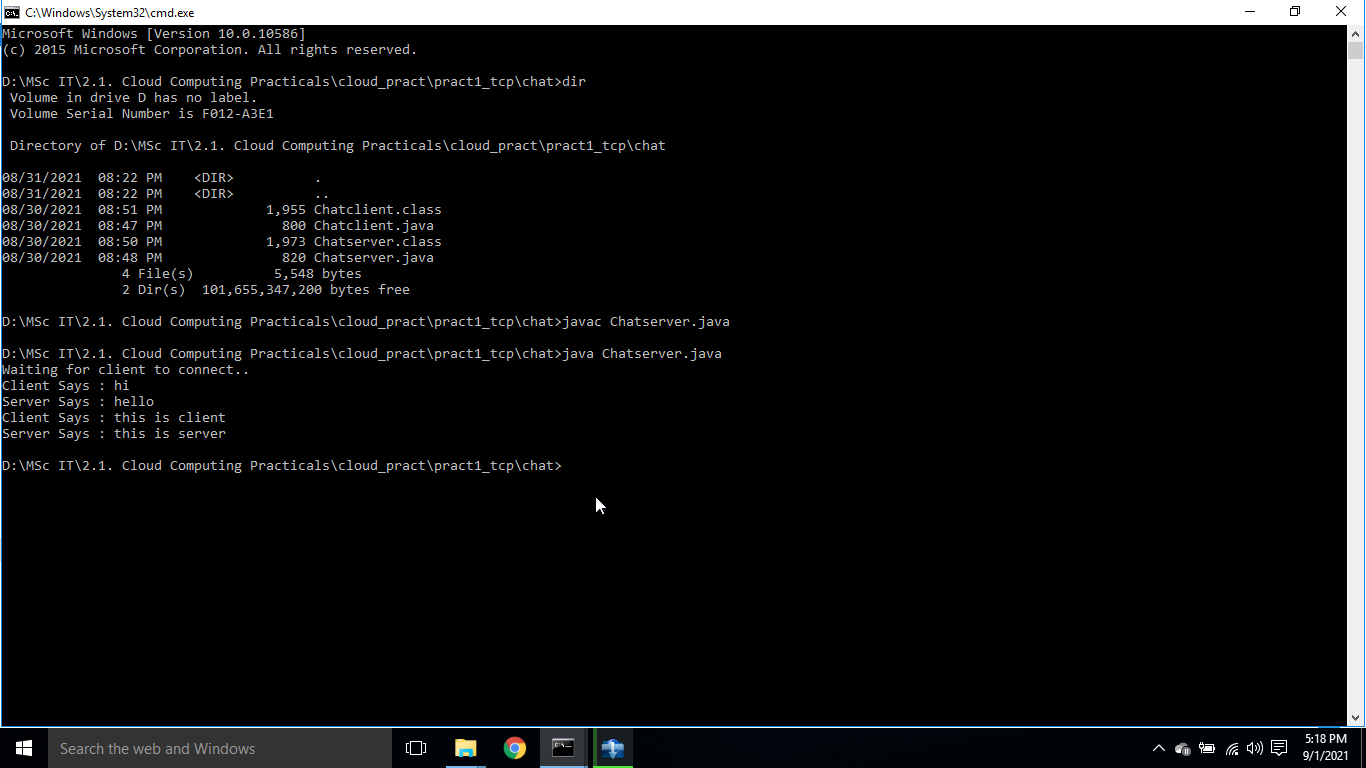
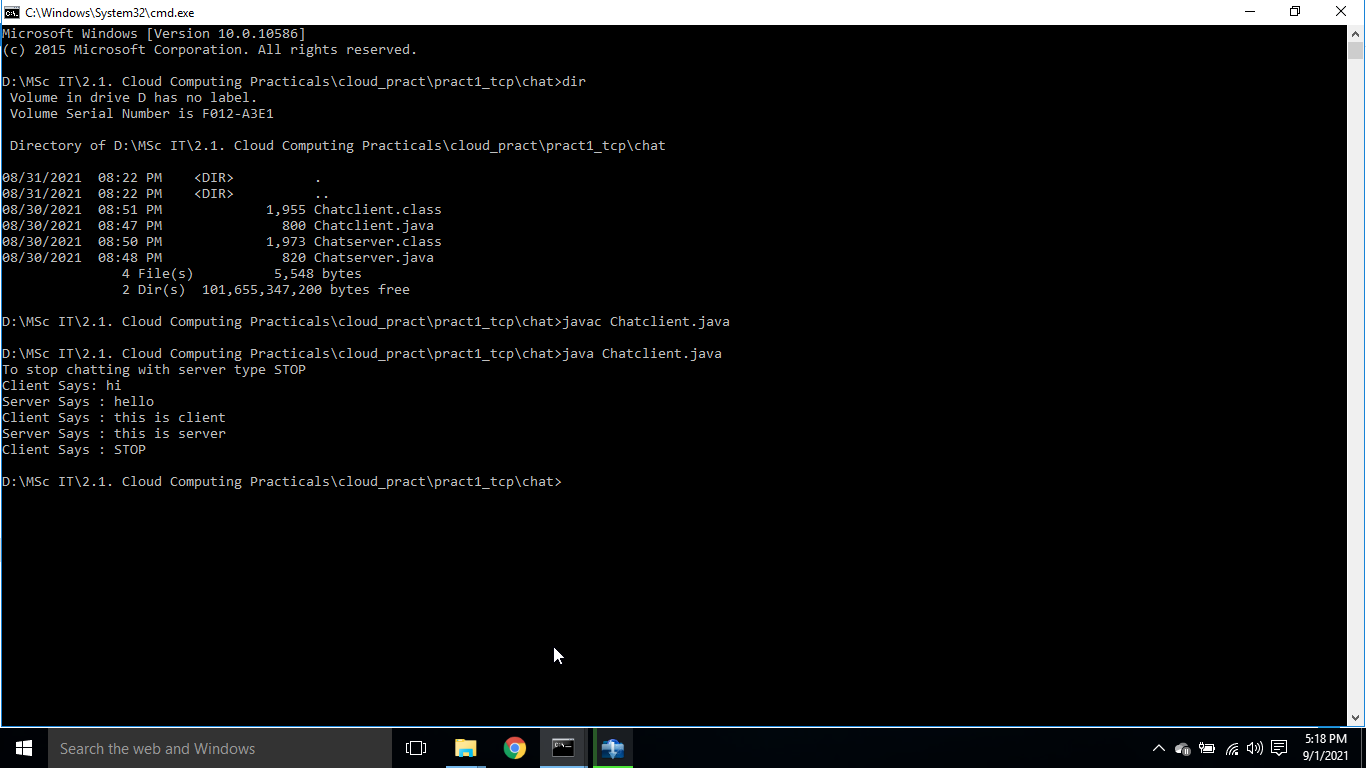
e.printStackTrace();

}

}

}

**Output:**



**Question No. 3 calculate square, squareroot, cube and cuberoot**

**Client Program:**

import java.io.\*;

import java.net.\*;

class clientCalc

{

clientCalc()

{

try

{

InetAddress ia = InetAddress.getLocalHost();

DatagramSocket ds = new DatagramSocket();

DatagramSocket ds1 = new DatagramSocket(1300);

System.out.println("\nRPC Client\n");

System.out.println("sqr, sqrt, cube cubert\n");

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 clientCalc();

}

}

**Server Program:**

import java.util.\*;

import java.net.\*;

import java.io.\*;

class serverCalc

{

DatagramSocket ds;

DatagramPacket dp;

String str,methodName,result;

int val;

serverCalc()

{

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("sqr"))

{

result= "" + square(val);

}

else if(methodName.equalsIgnoreCase("sqrt"))

{

result= "" + squareroot(val);

}

else if(methodName.equalsIgnoreCase("cube"))

{

result= "" + cube(val);

}

else if(methodName.equalsIgnoreCase("cubert"))

{

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 serverCalc();

}

}

Output:

**Output:** 