**Program:**

**Remote.java**

import java.rmi.\*;

import java.net.\*;

import java.io.\*;

import java.rmi.registry.\*;

public class remote

{

public static void main(String a[]) throws Exception

{

InetAddress ip=InetAddress.getLocalHost();

rev\_in obj =new rev\_im();

LocateRegistry.createRegistry(1900);

Naming.rebind("rmi://localhost:1900/"+ip.getHostName(),obj);

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

}}

**Interface class:**

import java.rmi.\*;

public interface rev\_in extends java.rmi.Remote

{

public int reverse(int x) throws RemoteException;

}

**Implementation class:**

import java.rmi.server.\*;

import java.io.\*;

public class rev\_im extends UnicastRemoteObject implements rev\_in

{

public rev\_im() throws Exception

{

super();

}

public int reverse(int x)

{

int res = 0;

while(x!=0)

{

int d=x%10;

res=(res\*10)+d;

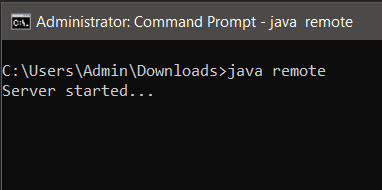
x=x/10;

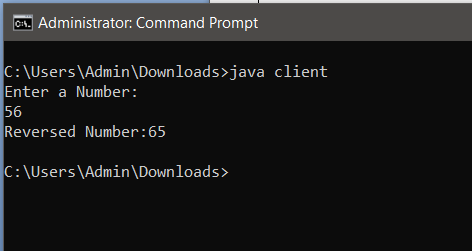
} return res;

}

}

**Output:**

****



**Program:**

**UPD server:**

import java.io.\*;

import java.net.\*;

public class udps

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

{ DatagramSocket ds = new DatagramSocket(8000);

byte[] receive = new byte[99999];

FileWriter fw=new FileWriter("udpin.txt");

while (true)

{

DatagramPacket DpReceive = new DatagramPacket(receive, receive.length);

ds.receive(DpReceive);

System.out.println("Client:-" + data(receive));

String ttt=data(receive).toString();

//fw.write(data(receive).toString());

for(int kk=0;kk<ttt.length();kk++)

fw.write(ttt.charAt(kk));

if (data(receive).toString().equals("exit"))

{

System.out.println("EXITING");

break;

}

receive = new byte[99999];

}

//System.out.println("EXITING");

}

public static StringBuilder data(byte[] a)

{

if (a == null)

return null;

StringBuilder ret = new StringBuilder();

int i = 0;

while (a[i] != 0)

{

ret.append((char) a[i]);

i++;

}

return ret;

}

}

**UDP client:**

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class udpcl

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

{ File f=new File("udpout.txt");

BufferedReader br=new BufferedReader(new FileReader(f));

DatagramSocket ds = new DatagramSocket();

InetAddress ip = InetAddress.getLocalHost();

byte buf[] = null;

//String inp;

while (true)

{

String inp = br.readLine();

buf = inp.getBytes();

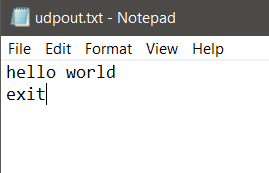
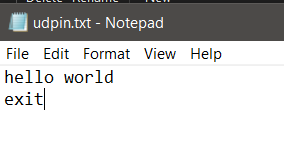
DatagramPacket DpSend =new DatagramPacket(buf, buf.length, ip, 8000);

ds.send(DpSend);

if (inp.equals("exit"))

break; }}}

**Output:**

****

**Program:**

**Client:**

import java.io.\*;

import java.net.\*;

public class tcpclient{

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

Socket socket =new Socket(InetAddress.getByName("LocalHost"),5000);

byte[] contents=new byte[10000];

//FileOutputStream fos=new FileOutputStream("/home/cse5lab/17c091/Desktop/helloworld.txt");

FileOutputStream fos=new FileOutputStream("tcpin.txt");

// FileOutputStream fos=new FileOutputStream("C:\Users\arun\Documents\nwlab\tcpin.txt");

BufferedOutputStream bos=new BufferedOutputStream(fos);

InputStream is=socket.getInputStream();

int bytesread=0;

while((bytesread=is.read(contents))!=-1)

bos.write(contents,0,bytesread);

bos.flush();

socket.close();

System.out.println("File saved successfully");

}}

**Server:**

import java.io.\*;

import java.net.\*;

public class tcpserver{

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

ServerSocket sock=new ServerSocket(5000);

Socket socket=sock.accept();

InetAddress ia=InetAddress.getByName("LocalHost");

//File file =new File("/home/cse5lab/17c091/Desktop/hello.txt");

File file =new File("tcpout.txt");

// FileOutputStream fos=new FileOutputStream("C:/Users/arun/Documents/nwlab/tcpout.txt");

FileInputStream fis=new FileInputStream(file);

BufferedInputStream bis=new BufferedInputStream(fis);

OutputStream os=socket.getOutputStream();

byte[] contents;

long filelength=file.length();

long current=0;

long start=System.nanoTime();

while(current!=filelength){

int size=10000;

if(filelength-current>=size)

current+=size;

else{

size=(int)(filelength-current);

current=filelength;

}

contents=new byte[size];

bis.read(contents,0,size);

os.write(contents);

System.out.println("sending file..."+(current+100)/filelength+"% complete");

}

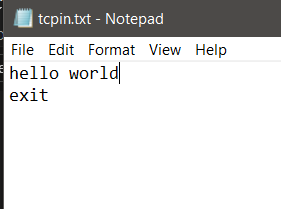
os.flush();

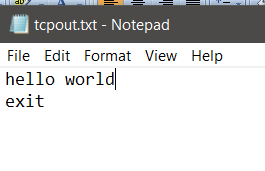
socket.close();

sock.close();

System.out.println("file sent successfully");}}

**Output:**

****

****

**Program:**

import java.net.\*;

import java.io.\*;

import java.util.\*;

public class dns1

{

public static void main(String args[])throws IOException

{

Process process = Runtime.getRuntime().exec("nslookup www.yahoo.com");

BufferedReader output = new BufferedReader(new InputStreamReader(process.getInputStream()));

String s=output.readLine();

System.out.println(s);

s=output.readLine();

System.out.println(s);

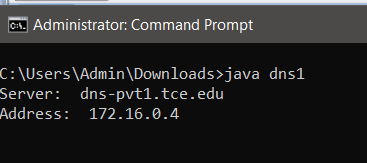
s=output.readLine();

System.out.println(s);

}

}

**Output:**



**Program:**

import java.net.\*;

import java.io.\*;

import java.util.\*;

public class dns2

{

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

Process process = Runtime.getRuntime().exec("nslookup www.yahoo.com");

BufferedReader output = new BufferedReader(new InputStreamReader(process.getInputStream()));

String s=output.readLine();

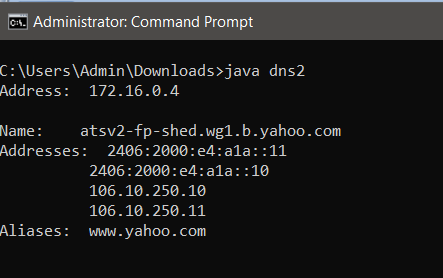
while((s=output.readLine())!=null)

System.out.println(s);

}

}

**Output:**

****