9.Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.

Server Program

```
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.net.ServerSocket;
import java.net.Socket;

public class SimpleFileServer {

  public final static int SOCKET_PORT = 13267;  // you may change this
  public final static String FILE_TO_SEND = "e:/source1.txt";  // you may change this

  public static void main (String [] args ) throws IOException {
    FileInputStream fis = null;
    BufferedInputStream bis = null;
    OutputStream os = null;
    ServerSocket servsock = null;
    Socket sock = null;
    try {
      servsock = new ServerSocket(SOCKET_PORT);
      while (true) {
        System.out.println("Waiting...");
        try {
          sock = servsock.accept();
          System.out.println("Accepted connection : " + sock);
          // send file
          File myFile = new File (FILE_TO_SEND);
          byte [] mybytearray  = new byte [(int)myFile.length()];
          fis = new FileInputStream(myFile);
          bis = new BufferedInputStream(fis);
          bis.read(mybytearray,0,mybytearray.length);
          os = sock.getOutputStream();
          System.out.println("Sending " + FILE_TO_SEND + "(" + mybytearray.length + " bytes)");
          os.write(mybytearray,0,mybytearray.length);
          os.flush();
          System.out.println("Done.");
```

```
      }
      finally {
        if (bis != null) bis.close();
        if (os != null) os.close();
        if (sock!=null) sock.close();
      }
    }
  }
  finally {
    if (servsock != null) servsock.close();
  }
 }
}
```

Client Program

```
import java.io.BufferedOutputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.net.Socket;


public class SimpleFileClient {

  public final static int SOCKET_PORT = 13267;      // you may change this

  public final static String SERVER = "127.0.0.1";  // localhost

  public final static String

      FILE_TO_RECEIVED = "e:/source-downloaded.txt";  // you may change this, I give a

                                                      // different name because i don't want to

                                                      // overwrite the one used by server...


  public final static int FILE_SIZE = 6022386; // file size temporary hard coded

                                  // should bigger than the file to be downloaded
```

```
  public static void main (String [] args ) throws IOException {
    int bytesRead;
    int current = 0;
    FileOutputStream fos = null
BufferedOutputStream bos = null;
    Socket sock = null;
    try {
      sock = new Socket(SERVER, SOCKET_PORT);
      System.out.println("Connecting...");
      // receive file
      byte [] mybytearray  = new byte [FILE_SIZE];
      InputStream is = sock.getInputStream();
      fos = new FileOutputStream(FILE_TO_RECEIVED);
      bos = new BufferedOutputStream(fos);
      bytesRead = is.read(mybytearray,0,mybytearray.length);
      current = bytesRead;
      do {
        bytesRead =
           is.read(mybytearray, current, (mybytearray.length-current));
        if(bytesRead >= 0) current += bytesRead;
      } while(bytesRead > -1);
      bos.write(mybytearray, 0 , current);
      bos.flush();
      System.out.println("File " + FILE_TO_RECEIVED
          + " downloaded (" + current + " bytes read)");
```

```
    }
    finally {
      if (fos != null) fos.close();
  if (bos != null) bos.close();
      if (sock != null) sock.close();
    }
  }
}
```

10. Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.

UDP Client

```
import java.io.*;
import java.net.*;
public class UDPC
{
public static void main(String[] args)
{
DatagramSocket skt;
 try {
skt=new DatagramSocket(); String msg= "text message "; byte[] b = msg.getBytes();
InetAddress host=InetAddress.getByName("127.0.0.1"); int serverSocket=6788;
DatagramPacket request =new DatagramPacket (b,b.length,host,serverSocket); skt.send(request);
byte[] buffer =new byte[1000];
DatagramPacket reply= new DatagramPacket(buffer,buffer.length); skt.receive(reply);
System.out.println("client received:" +new String(reply.getData())); skt.close();
}
catch(Exception ex)
{
}
```

```
}
}
```

UDP Server

```
import java.io.*; import java.net.*;
public class UDPS
{
public static void main(String[] args)
{
DatagramSocket skt=null;
try
{
skt=new DatagramSocket(6788); byte[] buffer = new byte[1000];
while(true)
{
DatagramPacket request = new DatagramPacket(buffer,buffer.length);
 skt.receive(request);
String[] message = (new String(request.getData())).split("");
 byte[] sendMsg= (message[1]+ " server processed").getBytes();
DatagramPacket reply = new
DatagramPacket(sendMsg,sendMsg.length,request.getAddress(),request.getPort());
skt.send(reply);
}
}
catch(Exception ex)
{
}
}
}
```