```
dout.writeUTF(s1);
dout.flush();
Sop("client");
}
catch(Excepn e) {
}
}
}
```

Program (MyClient1.java) → Multi Threaded client

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

public class MyClient1
 {
  Socket s;
  DataI/PStream din;
  DataO/PStream dout;
  public MyClient1()
  {
   try
    {
     s = new Socket ("localHost", 10);
     din = new DataInputStream (s.getInput-
                                  Stream());
     dout = new DataO/PStream (s.getop-
                                 -Stream());
     clientChat();
  }
  catch (Excep^n e) {Sop(e);}
}
public void clientChat() throws IoExcep^n
   {
    My m = new My(din);
    Thread t1 = new Thread(m);
     t1.start();
      BufferedReader br = new BufferedReader
          (new InputStreamReader (System.in));
```

```
String s1;
do
{
s1 = br.readLine();
dout.writeUTF(s1);
dout.flush();
}while(!s1.equals("stop"));
}

psvm()
{
new MyClient1();
}
}

class My implements Runnable
{
DataInputStream din;

My(DataInputStream din)
{
this.din = din;
}

public void run()
{
String s2="";
do
{
try{
```

```
            s2 = din.readUTF();
            Sop(s2);
        }
        catch(Exception e){}
    } while(!s2.equals("stop"));

    }

}
```

* Until here, we ~~transfer~~ sent/receive data with TCP protocol.

Assignment

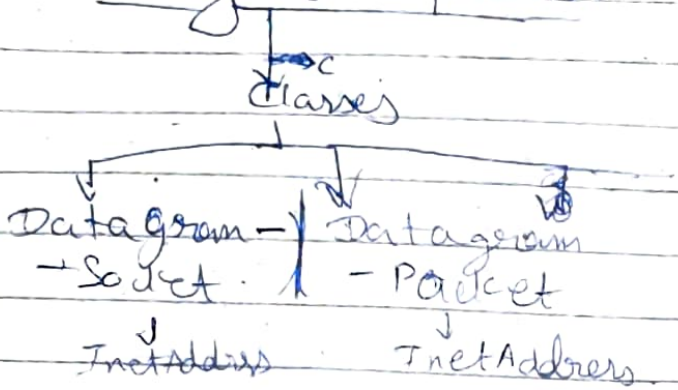
1. GUI based ~~Ether~~ Broadcast Chat

UDP Protocol

- Here, we just create a packet of data & sends it and no info. on successful delivery is received.

Now,

Using UDP protocol



InetAddress (Obj. of this class represents the IP address & name of the computer)

```
InetAddress I = InetAddress.getByName("Node10");
```
gives IP address of any node

or

```
InetAddress I = InetAddress.getLocalHost();
```
IP address of local host

```
String add = I.getHostAddress();
```
gives string of IP address

```
String name = I.getHostName();
```
gives string of computer name

Program (UdpServer.java)

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

class UdpServer
{
    psvm() throws Excepⁿ
    {
DatagramSocket ds = new DatagramSocket(8);
    byte b[] = new byte[24];
DatagramPacket dp = new DatagramPacket(b, b.length);
    ds.receive(dp);
    String s = new String(dp.getData());
        Sop(s.trim());
    }
}
```

→ PortNo.

→ removes whitespaces

Program (UdpClient.java)

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

class UdpClient
{
Public static void main() throws
                              Excep^n
 {
 DatagramSocket ds = new Datagram-
                          -Socket();
 String s = "india is a good country";
   byte b[] = s.getByte();
InetAddress i = InetAddress.getLocalHost();
DatagramPacket dp = new DatagramPacket
                          (b, b.length, i, 8);
   ds.send(dp);
 }
}
```

URL Class

↳ used to connect standalone Java prog to web based Applicⁿ

↳ eg. → during updatⁿ of Antivirus we connect to web based Applicⁿ.

Two classes → URLReader (only read)
→ URLConnecⁿReader (Read & write)

Program (URLReader.java) → connecting our program to Google's homepage & downloading it.

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

public class URLReader
{
    psvm() throws Excepⁿ {
    URL google = new URL("http://www.google.com/");
```

↓ slash means by default connect to home page

```
    BufferedReader in = new BufferedReader
        (new InputStreamReader
            (google.openStream()));

    String inputLine;
```

```
while((inputLine = in.readLine()) != null)
    Sop(inputLine);
    .
    in.close();
  }
}
```

Program (URLConnectionReader.java)

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

public class URLConnectionReader {
  static String inputLine = "";

psvm() throws Excepn {

URL google = new URL("http://www.google.
                      com/");

URL connecn yc = google.openConnection();
                 ↳ makes connctn // gives obj of URLConnecn

BufferedReader in = new BufferedReader(
                    new InputStreamReader(
                    yc.getInputStream()));
```

```
while(inputLine != null)
  {
  inputLine = in.readLine();
   Sop(inputLine);
  }
  in.close();
  }
}
```

On Cmd: D:\> javac We, > abc.html (Redirect)

Difference b/w URLConnection & URLReader.

→ URLConnection Reader can both read & write
→ But URLReader can only read.

Perl in java

FileExp with FTP

FileTransfer Protocol

Program (FileTransfer.java) [This is only for text file]

```
import java.awt.*;
import java.awt.event.*;
import java.javax.swing.*;
import java.io.*;
import java.net.*;

public class FileTransfer extends
        JFrame implements ActionListener
{
JFrame jf;
JButton jb1, jb2;
TextField tf;
JFileChooser jfc;
Socket s;
DataI/PStream dis;
DataO/PStream dout, dout1;
String ss1 = new String();
String ss2 = "";
File f;

FileTransfer()
{
jf = new JFrame("FILE TRANSFER");
jf.setSize(900, 900);
Container c = jf.getContentPane();
```

```
c.setBackground (Color.red);
jf.setLayout(null);
jb1 = new JButton("choose file");
jb2 = new JButton("send");
jb1.setBounds(30,50,100,50);
jb2.setBounds(250,150,70,50);
jf.add(jb1);
jf.add(jb2);
tf = new TextField();
tf.setEditable(false); → so no one can type
tf.setBackground(Color.white);
tf.setBounds(150,50,190,50);
jf.add(tf);
jf.setDefaultCloseOperation
        (WindowConstants.EXIT_ON_CLOSE);
jf.setVisible(true);
jfc = new JFileChooser();
jb1.addActionListener(this);
jb2 = addActionListener(this);
}
public void fileTransfer(String s1)
{
try
{
dout1.writeUTF(s1);
dout1.flush();
s2 = f.getAbsolutePath();
FileReader fr = new FileReader(s2);
```

```
BufferedReader br = new
                    BufferedReader(ps);
String ss2="";

do
{
ss=br.readLine()
 if (ss!=null)
 {
 dout.writeUTF(ss3);
 dout.flush();
 }
}while(ss3!=null);
}

Catch(IException e)
{
Sop(e+"jjj");
}
}
public void actionPerformed(ActionEvent e)
  {
    if(e.getSource()==jb1)
  {
    int x=jfc.ShowOpenDialog(null);
    if(x==JFileChooser.APPROVE_OPTION)
    {

      f=jfc.getSelectedFile();
    String path=f.getPath();
      ss1=f.getName();
```

```
        s1.tf.setText(path + "\\" + s1);
    }
}

if (e.getSource()==jb2)
{
 try
 {
//s1=tf.getText();
  s=new Socket("localHost",10);
//s =new Socket("54.---",10);
                IP.
 dout1 = new DataOPStream(s.getOutputStream());

dout = new DataOPstream(s.getOPStream());
}
Catch (Exception E)
 {
Sop("send button: "+E);
}
fileTransfer(s1);
 }
}
public static void main (String args[])
 {
 FileTransfer ft= new FileTransfer();
 }
}
```

(A)

Program (FileReceiver.java)

```
import java.awt.*;
import javax.swing.*;
import java.io.*;
impat java.net.*

public class FileReceiver
{
Socket s;
ServerSocket ss;
DataInputStream dis, dis1;
PrintWriter pw;

Pablic FileReceiver()
{
try
{
ss = new ServerSocket(10);
s = ss.accept();
dis = new DataInputStream(s.getInputStre-
                          -am());

String s2 = dis.readUTF();
FileWriter fo = new FileWriter(s2);
pw = new PrintWriter(fo);
fileReceiver();
}
```

difference doc^n in b/w C & Java

microsoft C came advance.

```
catch(Excep^n e)
 {
 Sop(e);
 }
}
public void fileReciver()
{
string str="";
try
{
do
{
str=dis.readUTF();
 pw.println(str);
 pw.flush();
}while(!str.equals(null));
}
catch(Excep^n e)
{
Sop(e)
}
}

 psvm(String args[])
 {
  new FileReceiver();
  }
 }
```

14/04/2018

Connectionless

Travelling Object on Network (via Serialization)

Program (emp.java) → we travel object of this class on the network

```
import java.io.*;

public class emp implements Serializable
{
String s;
int age;
emp(int age, String s)
  {
  this.s = s;
  this.age = age;
  }
void show()
  {
  Sop(s+age);
  }
}
```

Program (MyServer.java)

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

Public class MyServer
{
ServerSocket ss;
Socket s;
ObjectInputStream dis;

public MyServer()
{
try
{
Sop("ServerStarted");
ss = new ServerSocket(8);
s = ss.accept();
Sop("Client Connected");
dis = new ObjectInputStream(s.getInputstream());

emp z = (emp) dis.readObject();

z.show();
}
catch(Excepn e)
{
Sop(e);
}
}
```

psvm
{
new MyServer();
}
}

Program (MyClient.java)

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

public class MyClient1
{
Socket s;
ObjectOPStream dout;
public MyClient1()
{
try
{
s = new Socket("localhost", 8);
//s = new Socket("192.....", 8);
emp.e1 = new emp(10, "aaa");
dout = new ObjectOPStream
(s.getOPStream());
dout.writeObject(e1);
dout.flush();
}
catch(Exception e)
{sop(e);}
}
```

```
psvm( )
{
new MyClient1( )
}
}
```