NIELIT Gorakhpur

Course Name: A Level (2nd Sem) **Subject: JAVA**
Topic: Socket Programming(part 3) **Date: 20-05-20**

Java URLConnection class:

The URLConnection class is used for accessing the attribute of remote resource. URLConnection is the superclass of all the classes that represent a communication link between application and a URL.

Example:

```
import java.io.*;
import java.net.*;
public class UC {
public static void main(String[] args){
try
{
URL url=new URL("http://nielit.gov.in/gorakhpur");

URLConnection urlcon=url.openConnection();


InputStream stream=urlcon.getInputStream();
int i;
while((i=stream.read())!=-1)
{
System.out.print((char)i);
}
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

Returns a URLConnection instance that represents a connection to the remote

The getInputStream() method returns all the data of the specified URL

Note:

With the help of above program we can display the source code of a given page.

Client-server program in Java:

For this we need to write two programs one for client and other for server.

CLIENT PROGRAM:

```
import java.io.InputStream;
import java.io.DataInputStream;
import java.net.Socket;
public class ClientDemo
{
   public static void main(String args[])
{
       try
       {
          Socket s = new Socket("localhost", 7777);
          InputStream in = s.getInputStream();
          DataInputStream dis = new DataInputStream(in);
          String msg = dis.readLine();
          System.out.println("Server message is: "+ msg);
          dis.close();
          s.close();
       }
       catch(Exception e)
       {
          e.printStackTrace();
       }
 }
}
```

SERVER PROGRAM:

```
 import java.io.OutputStream;
import java.io.DataOutputStream;
import java.net.Socket;
import java.net.ServerSocket;
public class ServerDemo
{
    public static void main(String args[])
    {
        try
        {
            ServerSocket ss = new ServerSocket(7777);
            System.out.println("Server is ready.");
            Socket s = ss.accept();

          System.out.println("Connection created.");
            System.out.println("Sent message to client.");
            OutputStream out = s.getOutputStream();
            DataOutputStream dos = new DataOutputStream(out);
            dos.writeBytes("Welcome");
            dos.close();
            s.close();
            ss.close();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Listens for a connection to be made to this socket and

Explanation:

Write the programs then open two command prompts and compile both after that first run server program and after that client program, the message from the server will be printed on client command prompt. Note that the port number must be same on client as well as on server.