

Practical No. 2

Aim :- Create, debug and execute program based on TCP client server communication.

Theory :-

To examine a socket at anytime for address and port information by using following methods.

① `InetAddress getAddress()`

Returns `InetAddress` associated with socket IP address.

② `int getPort()`

Returns the remote port which socket object is connected.

③ `int getLocalhost()`

Returns the local port to which socket object is connected.

④ `InputStream getInputStream()`

Returns the input stream associated with invoking socket.

⑤ `OutputStream getOutputStream()`

Returns the output stream associated with invoking socket.

Establishing a TCP connection using TCP socket.

- ① Server need to instantiate server socket object
- ② Server invokes an `accept()` by server socket class
- ③ client \rightarrow socket object which has server name & port number
- ④ Attempt by client to connect

Some methods used in server sockets

i `accept()` - it listen for connection & accepts it.

ii `close()` - closes this socket

iii `getLocalPort()` - returns port no. on which this socket is listening

Program:-

client.java 1 x

client.java > client > main(String[])

```
1  import java.net.*;
2  import java.io.*;
3
4  public class client {
    Run | Debug
5      public static void main(String[] args) throws IOException {
6          Socket s = new Socket("localhost", 4999);
7
8          PrintWriter pr = new PrintWriter(s.getOutputStream());
9          pr.println("Is it working?");
10         pr.flush();
11
12         InputStreamReader in = new InputStreamReader(s.getInputStream());
13         BufferedReader bf = new BufferedReader(in);
14
15         String str = bf.readLine();
16         System.out.println("Server:"+str);
17     }
18 }
```


server.java 1 x

server.java > server > main(String[])

```
1  import java.net.*;
2  import java.io.*;
3
4  public class server {
    Run | Debug
5      public static void main(String[] args) throws IOException {
6          ServerSocket ss = new ServerSocket(4999);
7          Socket s = ss.accept();
8
9          System.out.println("Client is Connected");
10
11         InputStreamReader in = new InputStreamReader(s.getInputStream());
12         BufferedReader bf = new BufferedReader(in);
13
14         String str = bf.readLine();
15         System.out.println("Client:"+str);
16
17         PrintWriter pr = new PrintWriter(s.getOutputStream());
18         pr.println("Yess!");
19         pr.flush();
20     }
21 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

F:\BHT-TK-001\With Sem\IT501E - Advanced Java\Practicals\Practical No.2>javac client.java

F:\BHT-TK-001\With Sem\IT501E - Advanced Java\Practicals\Practical No.2>java client
Server:Yess!

F:\BHT-TK-001\With Sem\IT501E - Advanced Java\Practicals\Practical No.2>

Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

F:\BHT-TK-001\With Sem\IT501E - Advanced Java\Practicals\Practical No.2>javac server.java

F:\BHT-TK-001\With Sem\IT501E - Advanced Java\Practicals\Practical No.2>java server
Client is Connected
Client:Is it working?

F:\BHT-TK-001\With Sem\IT501E - Advanced Java\Practicals\Practical No.2>