Finolex Academy of Management and Technology, Ratnagiri

Department of Information Technology

Subject:	**Networking Lab (ITL401)**		
Class:	**SE IT / Semester – IV (CBCGS) / Academic year: 2017-18**		
Name of Student:	**Kazi Jawwad A Rahim**		
Roll No:	**28**	**Date of performance (DOP) :**	
Experiment No:	**11**	**Date of checking (DOC) :**	
Title: To implement Socket Programming with Java: TCP Client, TCP Server.			
Marks:		**Teacher's Signature:**	

1. Aim: To implement Socket Programming with Java: TCP Client, TCP Server.

2. Prerequisites:

Knowledge of

1. Java programming
2. TCP/IP

3. Hardware Requirements:

1. PC with minimum 2GB RAM

4. Software Requirements:

1. Linux (Ubuntu 10.04)/ Windows
2. Jdk installed

5. Learning Objectives:

1. To understand basic concepts of Socket Programming.
2. To be able to implement client server programming in java.
3. To understand basic java.net package features.

6. Course Objectives Applicable: LO 5

7. Program Outcomes Applicable: PO2, PO4

8. Program Education Objectives Applicable: 1, 3

TCP server code:

```
import java.io.*;
import java.net.*;
class TCPServer
{
public static void main(String argv[]) throws Exception
{
String clientSentence;
String capitalizedSentence;
ServerSocket welcomeSocket = new ServerSocket(6789);
while(true)
{
Socket connectionSocket = welcomeSocket.accept();
BufferedReader inFromClient = new BufferedReader(new
InputStreamReader(connectionSocket.getInputStream()));
DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());
clientSentence = inFromClient.readLine();
capitalizedSentence = clientSentence.toUpperCase() + '\n';
outToClient.writeBytes(capitalizedSentence);
}
}
}
```

TCP client code:

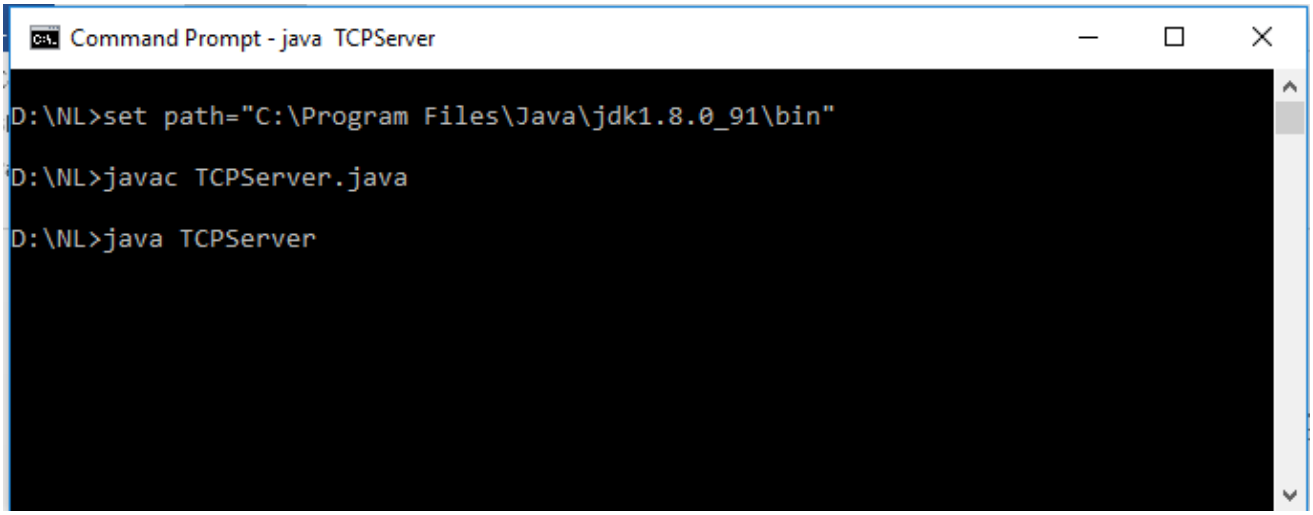
```
import java.io.*;

import java.net.*;

class TCPClient

{

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

{

String sentence;

String modifiedSentence;

BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));

Socket clientSocket = new Socket("localhost", 6789);

DataOutputStream outToServer =new DataOutputStream(clientSocket.getOutputStream());

BufferedReader inFromServer = new BufferedReader(new

InputStreamReader(clientSocket.getInputStream()));

sentence = inFromUser.readLine();

outToServer.writeBytes(sentence + '\n');
```

```
modifiedSentence = inFromServer.readLine();
System.out.println("FROM SERVER: " + modifiedSentence);
clientSocket.close();
}
}
```

To run on Terminal or Command Prompt

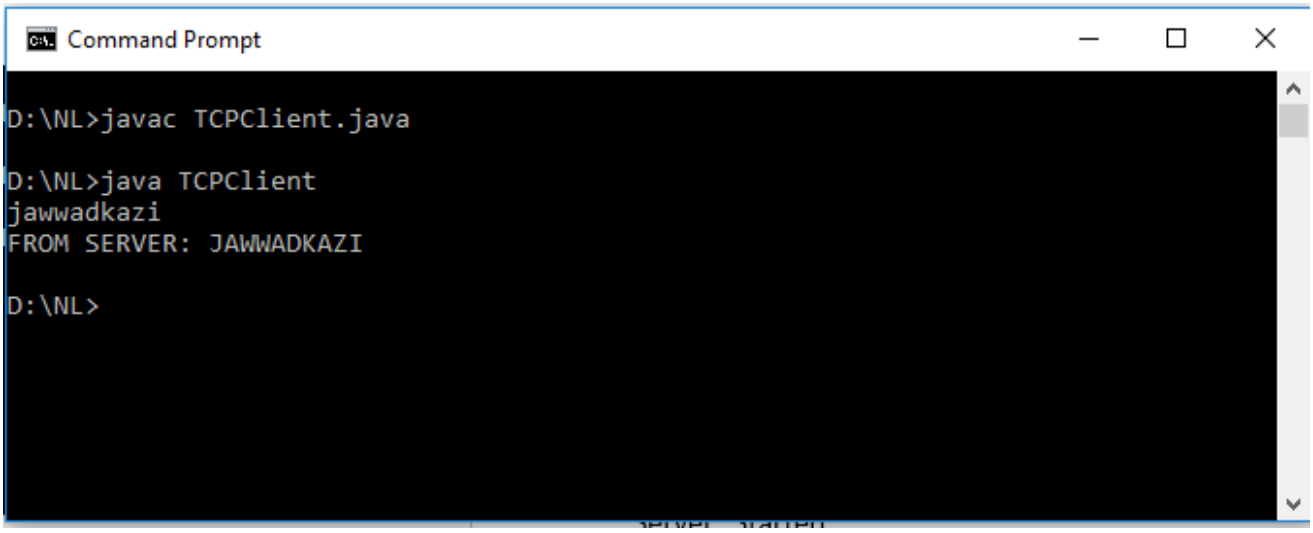
Open two windows one for Server and another for Client

1. First run the Server application as,



Server Started
Waiting for a client …

2. Then run the Client application on another terminal as,



13. Experiment/Assignment Evaluation

SR	Parameters	Weight	Excellent	Good	Average	Poor	Not as per requirement
		Scale Factor ->	5	4	3	2	0
1	Technical Understanding	25					
2	Performance / Execution	25					
3	Question Answers	20					
4	Punctuality	20					
5	Presentation	10					
	Total out of 100 --> #(to be converted as per term-work evaluation applicable to the subject)		**∑ (Weight * Scale Factor)/5 = _____**				

References:

[1] http://www.buyya.com/java/Chapter13.pdf
[2] http://www.scit.wlv.ac.uk/~in8297/CP4044/lectures/L07.pdf
[3] http://www.kiv.zcu.cz/~ledvina/Knihovnicka/Sockets_Java.pdf

Viva Questions

1. What is Socket?
2. Which classes of from java.net package is used in creation of server and client application?
3. What is well known ports?