TRINITY INTERNATIONAL COLLEGE

(Tribhuvan University Affiliated)

Lab Report:5 Network Programming

Submitted by:

Name :Anusha Panta
Program : **B. Sc. (CSIT)**
Roll No :10
Semester: 7th
Date : 21/06/2020

Submitted to:

Aman Maharjan

KATHMANDU, NEPAL
2020

Unit 5: Network Programming

1. **Write two programs that can communicate in a network using TCP Socket? [2070, 7073, 2074]**

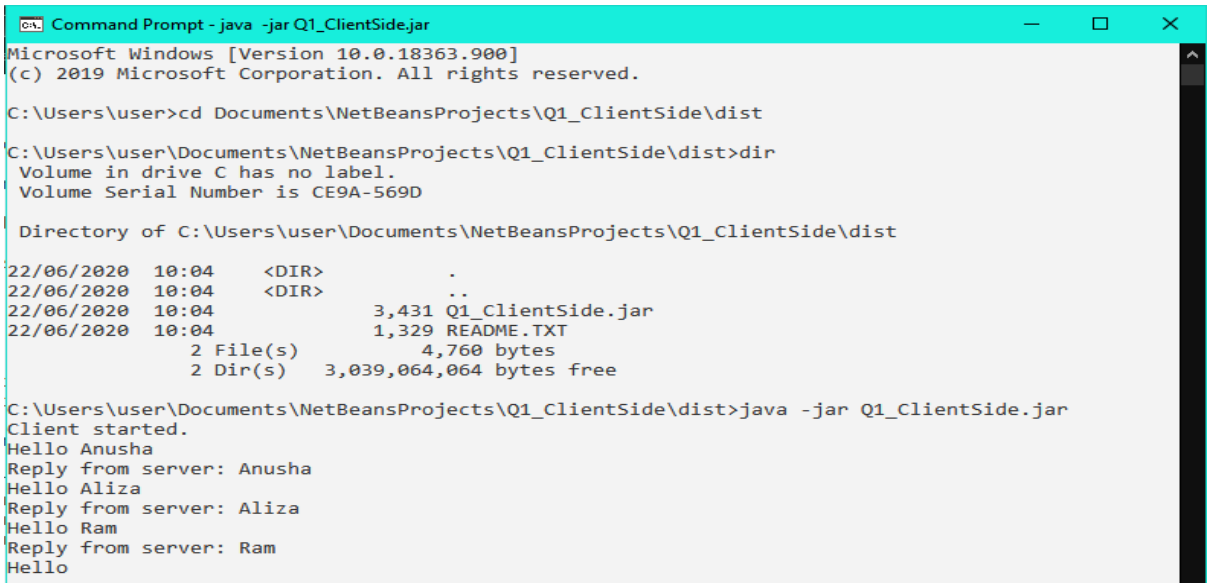
Q1_ClientSide.java

```
package q1_clientside;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

public class Q1_ClientSide {

    /**
     * @param args the command line arguments
     * @throws java.io.IOException
     */
    public static void main(String[] args) throws IOException {
        final String HOST = "127.0.0.1";
        final int PORT = 6789;

        System.out.println("Client started.");
        try (
                Socket socket = new Socket(HOST, PORT);
                PrintWriter out = new
                PrintWriter(socket.getOutputStream(), true);
                Scanner in = new Scanner(socket.getInputStream());
                Scanner s = new Scanner(System.in);
        ) {
            while (true) {
                System.out.print("Hello ");
                String input = s.nextLine();
                out.println(input);
                if (input.equalsIgnoreCase("exit")) break;

                System.out.println("Reply from server: " +
                in.nextLine());
            }
        }
    }

}
```

Q1_ServerSide.java

```
package q1_serverside;

import java.io.IOException;
```

```
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

public class Q1_ServerSide {

    /**
     * @param args the command line arguments
     * @throws java.io.IOException
     */
    public static void main(String[] args) throws IOException {
        final int PORT = 6789;
        System.out.println("Server started.");
        System.out.println("Listening to client...");

        try (
                ServerSocket serverSocket = new ServerSocket(PORT);
                Socket clientSocket = serverSocket.accept();
                PrintWriter out = new
PrintWriter(clientSocket.getOutputStream(), true);
                Scanner in = new
Scanner(clientSocket.getInputStream());
        ) {
            while (true) {
                String input = in.nextLine();
                if (input.equalsIgnoreCase("exit")) break;
                System.out.println("Received from client: " +
input);
                out.println(input);
            }
        }
        System.out.println("Server stopped");
    }
}
```



```
Command Prompt - java -jar Q1_ServerSide.jar
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\user>cd Documents\NetBeansProjects\Q1_ServerSide\dist

C:\Users\user\Documents\NetBeansProjects\Q1_ServerSide\dist>dir
 Volume in drive C has no label.
 Volume Serial Number is CE9A-569D

 Directory of C:\Users\user\Documents\NetBeansProjects\Q1_ServerSide\dist

22/06/2020  10:05    <DIR>          .
22/06/2020  10:05    <DIR>          ..
22/06/2020  10:05             3,450 Q1_ServerSide.jar
22/06/2020  10:05             1,329 README.TXT
               2 File(s)          4,779 bytes
               2 Dir(s)   3,038,666,752 bytes free

C:\Users\user\Documents\NetBeansProjects\Q1_ServerSide\dist>java -jar Q1_ServerSide.jar
Server started.
Listening to client...
Received from client: Anusha
Received from client: Aliza
Received from client: Ram
```

2. Write a program to illustrate the use of InetAddress class. [2073]

```
package labasssignment_5;

import java.net.InetAddress;

public class Q2_InetAddress {

public static void main(String[] args) {
try {
InetAddress ip = InetAddress.getByName("www.trinitycollege.edu.np");

System.out.println("Host Name: " + ip.getHostName());
System.out.println("IP Address: " + ip.getHostAddress());
} catch (Exception e) {
System.out.println(e);

}
}
}
```

```
Output - Q2_InetAddressClass (run)
run:
Host Name: www.trinitycollege.edu.np
IP Address: 202.52.234.236
BUILD SUCCESSFUL (total time: 1 second)
```

3. **Write client and server programs in which a server program accepts a radius of a circle from the client program, computes area, sends the computed area to the client program, and displays it by client program. [2075]**

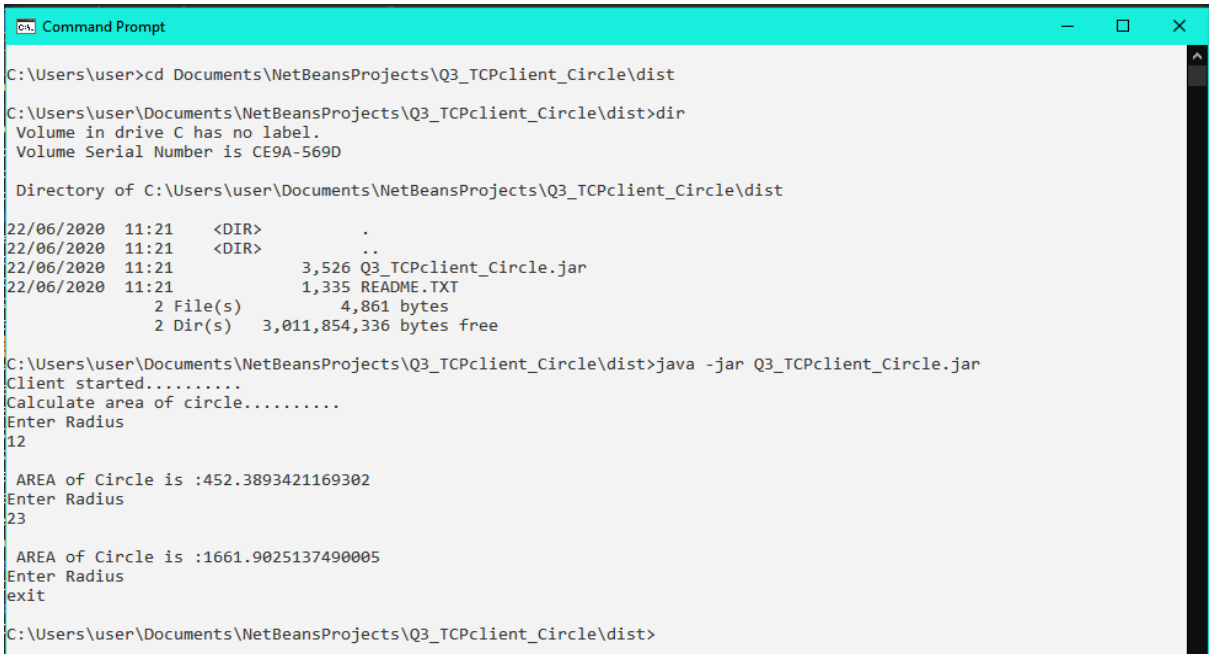
TCPclient_Circle.java

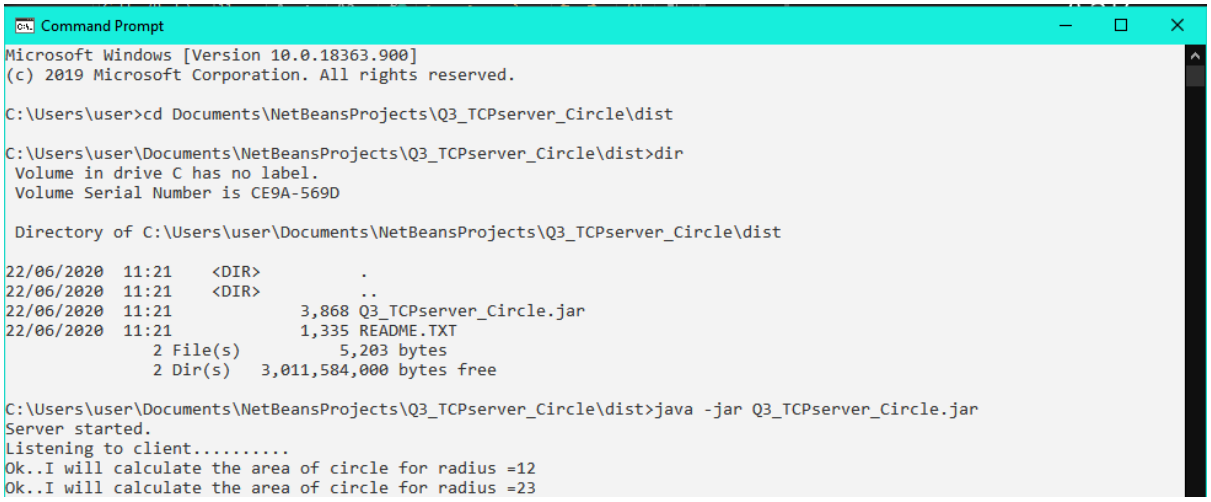
```
package tcpclient_circle;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

public class TCPclient_Circle {

public static void main(String[] args) throws IOException {
final String HOST = "127.0.0.1";
final int PORT = 1234;

System.out.println("Client started.........");
System.out.println("Calculate area of circle.........");

try (
Socket socket = new Socket(HOST, PORT);
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
Scanner in = new Scanner(socket.getInputStream());
Scanner s = new Scanner(System.in);) {
while (true) {
System.out.println("Enter Radius");
String radius = s.nextLine();
out.println(radius);
if (radius.equalsIgnoreCase("exit")) {
break;
}
System.out.println("\n AREA of Circle is :" + in.nextLine());
}
}
}
}
```

TCPserver_Circle.java

```
package tcpserver_circle;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;
```

```
public class TCPserver_Circle {

public static void main(String[] args) throws IOException {
final int PORT = 1234;
System.out.println("Server started.");
System.out.println("Listening to client.........");

try (
ServerSocket serverSocket = new ServerSocket(PORT);
Socket clientSocket = serverSocket.accept();
PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
Scanner in = new Scanner(clientSocket.getInputStream());) {
while (true) {
String inputline = in.nextLine();
int radius = Integer.valueOf(inputline);

if (inputline.equalsIgnoreCase("exit")) {

break;
}

System.out.println("Ok..I will calculate the area of circle for "
+ "radius =" + radius);

double area = Math.PI * radius * radius;
inputline = Double.toString(area);
out.println(inputline);
}
}
System.out.println("Server stopped.");
}
}
```

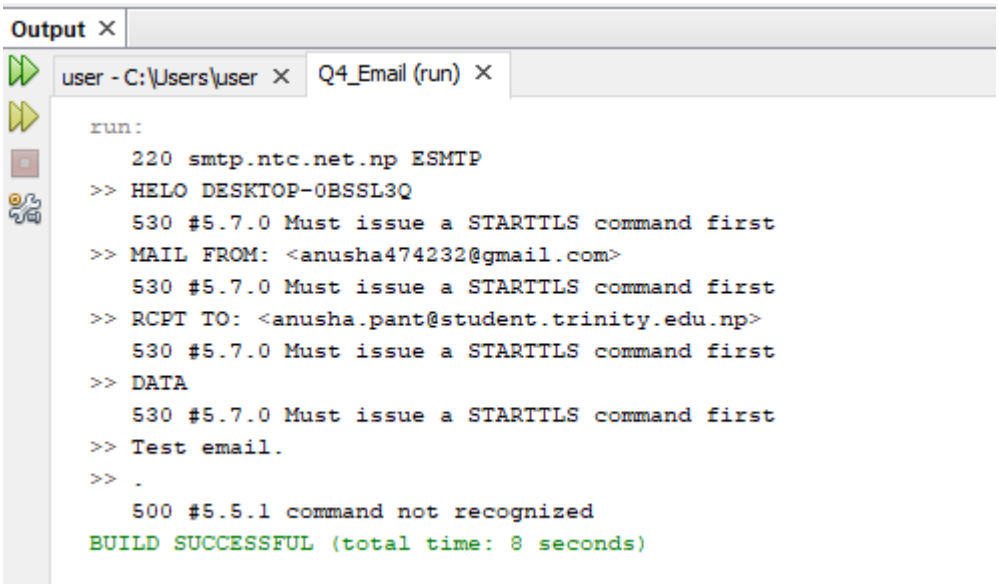




4. Write a program to send email using Java [2073, 2074]

```
package q4_email;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.util.Scanner;

public class Email_Send {

    public static void main(String[] args) throws IOException {
        Email email = new Email(
                "anusha474232@gmail.com",
                "anusha.pant@student.trinity.edu.np",
                "Test email.");
        email.send();
    }
}

class Email {
    private Scanner in = null;
    private PrintWriter out = null;
    private final String SMTP_SERVER = "smtp.ntc.net.np";
    private final int SMTP_PORT = 25;
    private String from = null;
    private String to = null;
    private String message = null;

    public Email(String from, String to, String message) {
        this.from = from;
        this.to = to;
        this.message = message;
```

```
    }

    private void send(String s) throws IOException {
        System.out.println(">> " + s);
        out.print(s.replaceAll("\n", "\r\n"));
        out.print("\r\n");
        out.flush();
    }

    private void receive() throws IOException {
        String line = in.nextLine();
        System.out.println("   " + line);
    }

    public void send() throws IOException {
        Socket socket = new Socket(SMTP_SERVER, SMTP_PORT);
        in = new Scanner(socket.getInputStream());
        out = new PrintWriter(socket.getOutputStream(), true);
        String hostName = InetAddress.getLocalHost().getHostName();
        receive();
        send("HELO " + hostName);
        receive();
        send("MAIL FROM: <" + from + ">");
        receive();
        send("RCPT TO: <" + to + ">");
        receive();
        send("DATA");
        receive();
        send(message);
        send(".");
        receive();
        socket.close();
    }
}
```



5. Write client and server programs in which a server program accepts the length and breadth of a rectangle from the client program, computes area, sends the computed area to the client program, and displays it by client program.

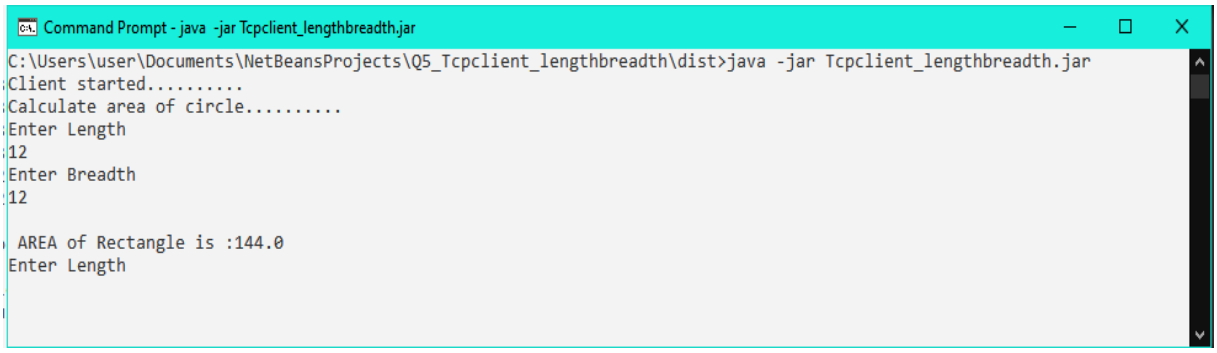
Tcpclient_lengthbreadth.java

```
package tcpclient_lengthbreadth;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

public class Tcpclient_lengthbreadth {

public static void main(String[] args) throws IOException {
final String HOST = "127.0.0.1";
final int PORT = 1234;

System.out.println("Client started.........");
System.out.println("Calculate area of circle.........");

try (
Socket socket = new Socket(HOST, PORT);
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
Scanner in = new Scanner(socket.getInputStream());
Scanner s = new Scanner(System.in);) {
while (true) {
System.out.println("Enter Length");
String length = s.nextLine();
out.println(length);
System.out.println("Enter Breadth");
String breadth = s.nextLine();
out.println(breadth);
if (length.equalsIgnoreCase("exit")) {
break;
}
System.out.println("\n AREA of Rectangle is :" + in.nextLine());
}
}
}

}
```

Tcpserver_lengthbreadth.java

```
package tcpserver_lengthbreadth;

import java.io.IOException;
import java.io.PrintWriter;
```

```
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

public class Tcpserver_lengthbreadth {

public static void main(String[] args) throws IOException {
final int PORT = 1234;
System.out.println("Server started.");
System.out.println("Listening to client..........");

try (
ServerSocket serverSocket = new ServerSocket(PORT);
Socket clientSocket = serverSocket.accept();
PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
Scanner in = new Scanner(clientSocket.getInputStream());) {
while (true) {
String line1 = in.nextLine();
int length = Integer.valueOf(line1);
String line2 = in.nextLine();
int breadth = Integer.valueOf(line2);

if (line1.equalsIgnoreCase("exit")) {

break;
}

System.out.println("Ok..I will calculate the area of rectangle for "
+ "length =" + length + "breadth="+breadth);

double area = length*breadth;
String inputline = Double.toString(area);

out.println(inputline);
}
}

System.out.println("Server stopped.");
}

}
```



6. Write echo server and echo client program using UDP.

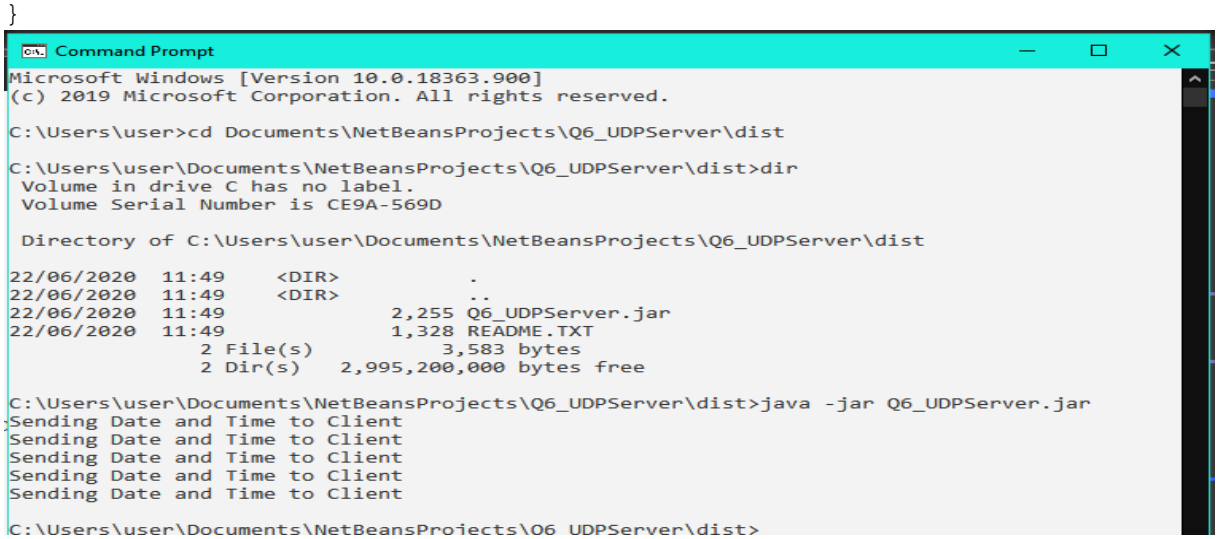
UDPClient.java

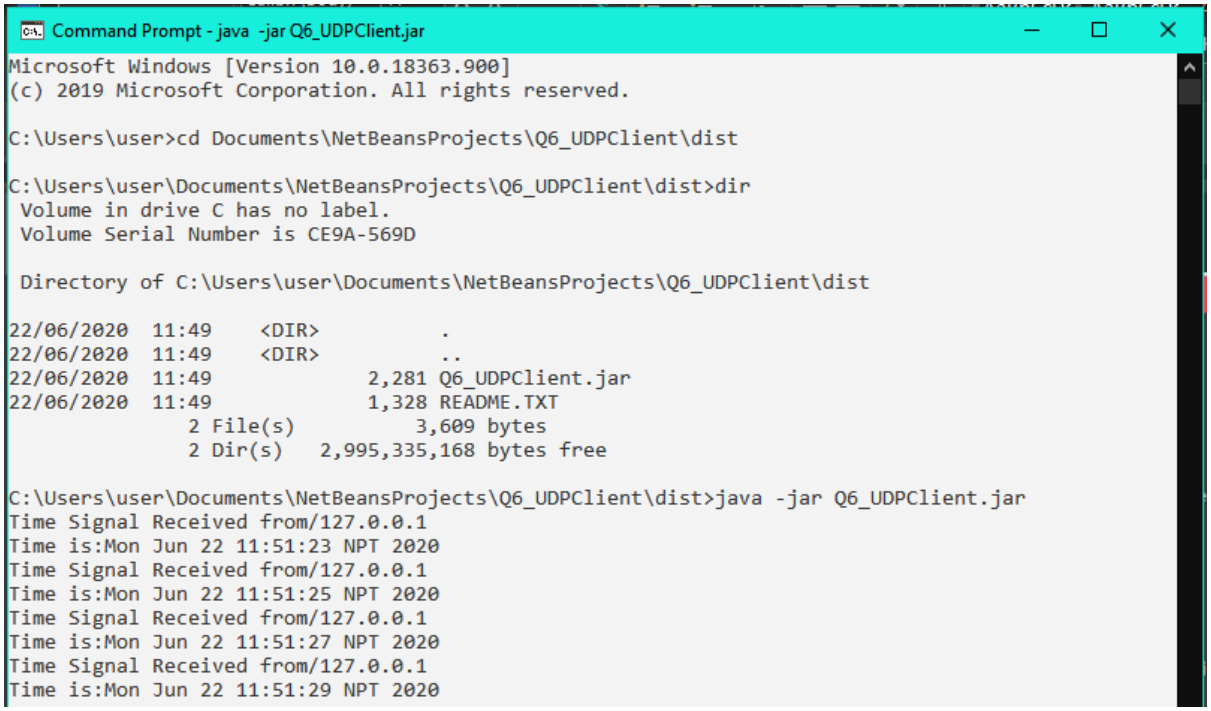
```
package udpclient;

import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class UDPClient {

DatagramSocket ds;
DatagramPacket dp;
public static void main(String[] args) {
UDPClient client=new UDPClient();
client.receiveDateTime();
}
public void receiveDateTime(){
byte b[]=new byte[64];
String str;
try{
ds=new DatagramSocket(1234);
dp=new DatagramPacket(b,b.length);
while(true){
ds.receive(dp);
str=new String(dp.getData());
System.out.println("Time Signal Received from"+dp.getAddress()+
"\nTime is:"+str);
}
}
catch(Exception e){
System.out.println(e);
}
```

```
}
}
```

UDPServer.java

```
package udpserver;

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Date;

public class UDPServer {

DatagramSocket ds;
DatagramPacket dp;
InetAddress addr;
public static void main(String[] args) throws Exception {
UDPServer db = new UDPServer();
db.sendDateTime();
}
public UDPServer() throws Exception{
addr=InetAddress.getByName("localhost");
ds=new DatagramSocket();
}
public void sendDateTime() throws Exception{
byte[] buff;
for(int i=1;i<=5;i++){
Thread.sleep(2000);
System.out.println("Sending Date and Time to Client");
String s=(new Date()).toString();
buff=s.getBytes();
dp=new DatagramPacket(buff,buff.length,addr,1234);
ds.send(dp);
}
}
}
```





7. Write client and server programs in which a server program accepts a radius of a circle from the client program, computes area, sends the computed area to the client program, and displays it by client program. The server should be able to handle multiple clients.

Thread_TCPcircle_client.java

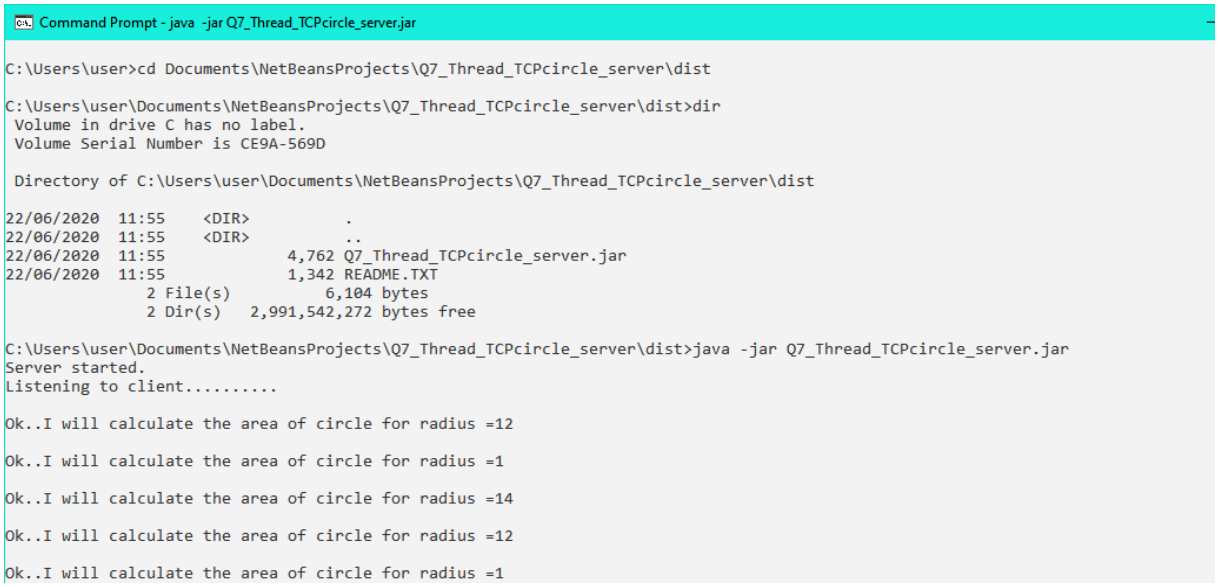
```
package thread_tcpcircle_client;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

public class Thread_TCPcircle_client {

public static void main(String[] args) throws IOException {
final String HOST = "127.0.0.1";
final int PORT = 1234;

System.out.println("Client started.....");
System.out.println("Calculate area of circle.....");

try (
Socket socket = new Socket(HOST, PORT);
```

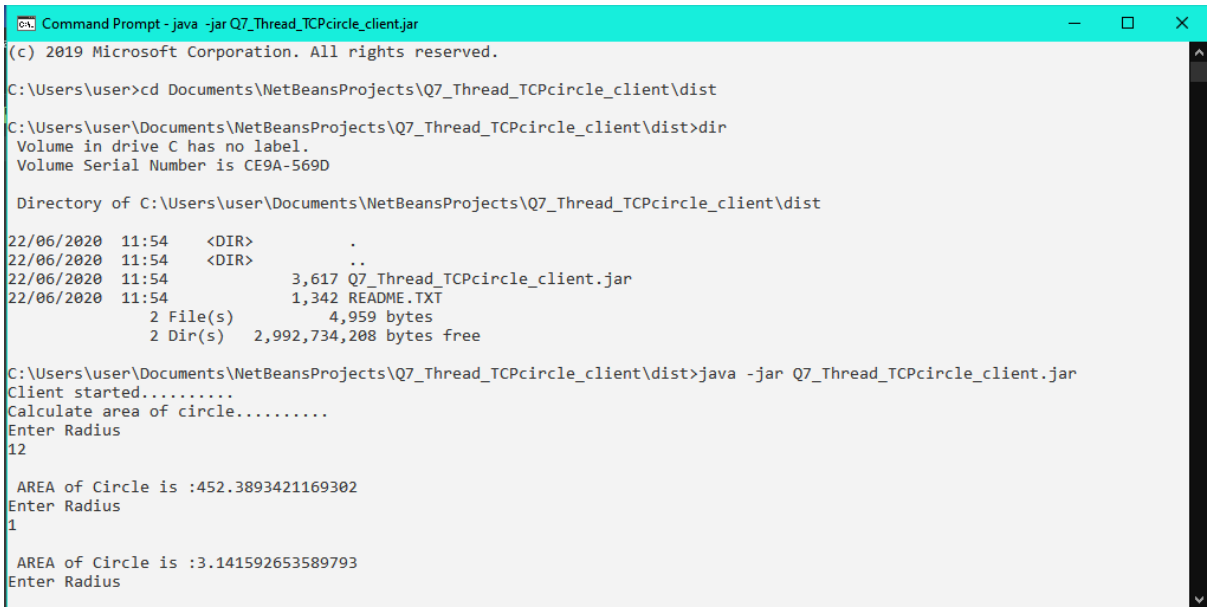
```
                PrintWriter out = new
                PrintWriter(socket.getOutputStream(), true);
Scanner in = new Scanner(socket.getInputStream());
Scanner s = new Scanner(System.in);) {
while (true) {
System.out.println("Enter Radius");
String radius = s.nextLine();
out.println(radius);
if (radius.equalsIgnoreCase("exit")) {
break;
}
System.out.println("\n AREA of Circle is :" +
in.nextLine());
}
}
}

}
```

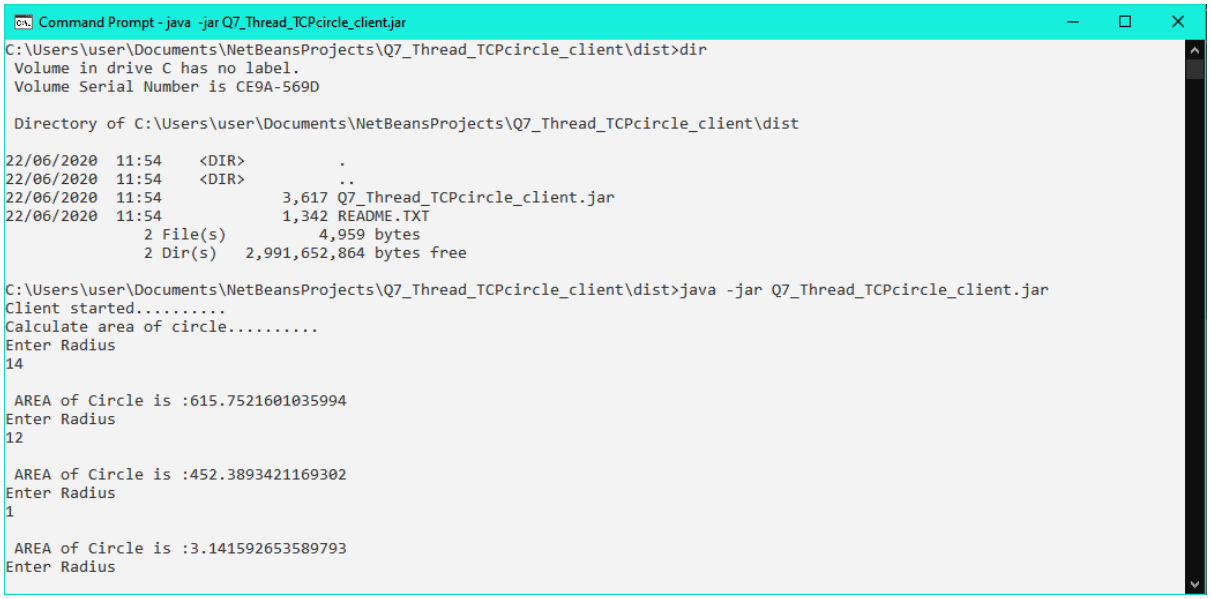
Thread_TCPcircle_server.java

```
package thread_tcpcircle_server;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

public class Thread_TCPcircle_server {

public static void main(String[] args) throws
IOException,NumberFormatException {
final int PORT = 1234;
ServerSocket serverSocket = new ServerSocket(PORT);

System.out.println("Server started.");
System.out.println("Listening to client.........");

while (true) {
Socket clientSocket = serverSocket.accept();
Thread t = new Thread() {
@Override
public void run() {
try (
PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
Scanner in = new Scanner(clientSocket.getInputStream());) {
while (in.hasNextLine()) {
String inputline = in.nextLine();
int radius = Integer.valueOf(inputline);
```

```
if (inputline.equalsIgnoreCase("exit")) {

break;
}
System.out.println("\nOk..I will calculate the area of circle for "
+ "radius =" + radius);
double area = Math.PI * radius * radius;
inputline = Double.toString(area);
out.println(inputline);
}
} catch (IOException e) {
}
}
};
t.start();
}
}
}
```







8. **Write client and server programs in which a server program accepts the length and breadth of a rectangle from the client program, computes area, sends the computed area to the client program, and displays it by client program. The server should be able to handle multiple clients.**

Thread_TCPrectangle_client.java

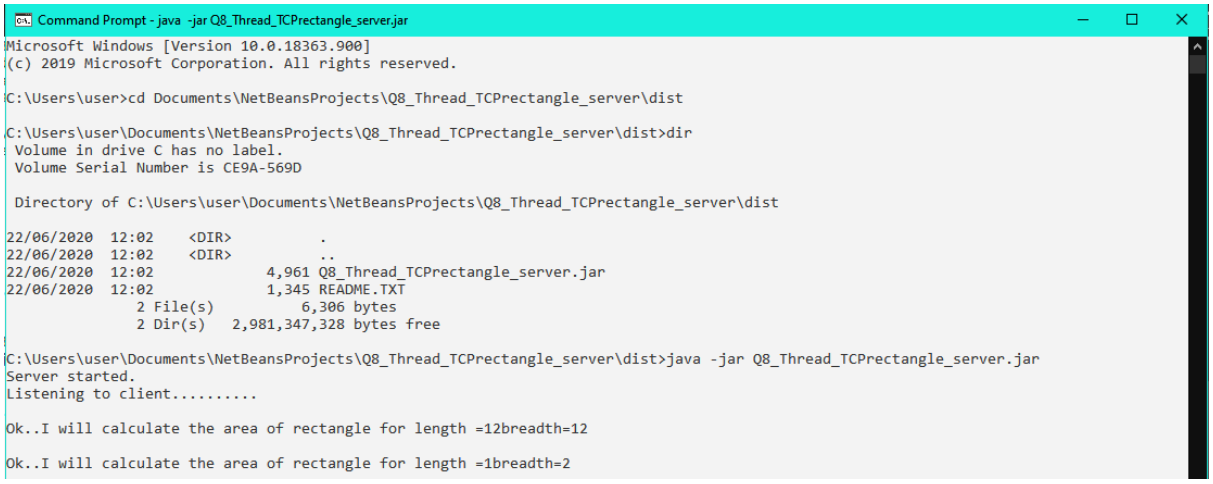
```
package thread_tcprectangle_client;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

public class Thread_TCPrectangle_client {

public static void main(String[] args) throws IOException {
final String HOST = "127.0.0.1";
final int PORT = 1234;

System.out.println("Client started.....");
System.out.println("Calculate area of rectangle.....");

try (
Socket socket = new Socket(HOST, PORT);
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
Scanner in = new Scanner(socket.getInputStream());
```

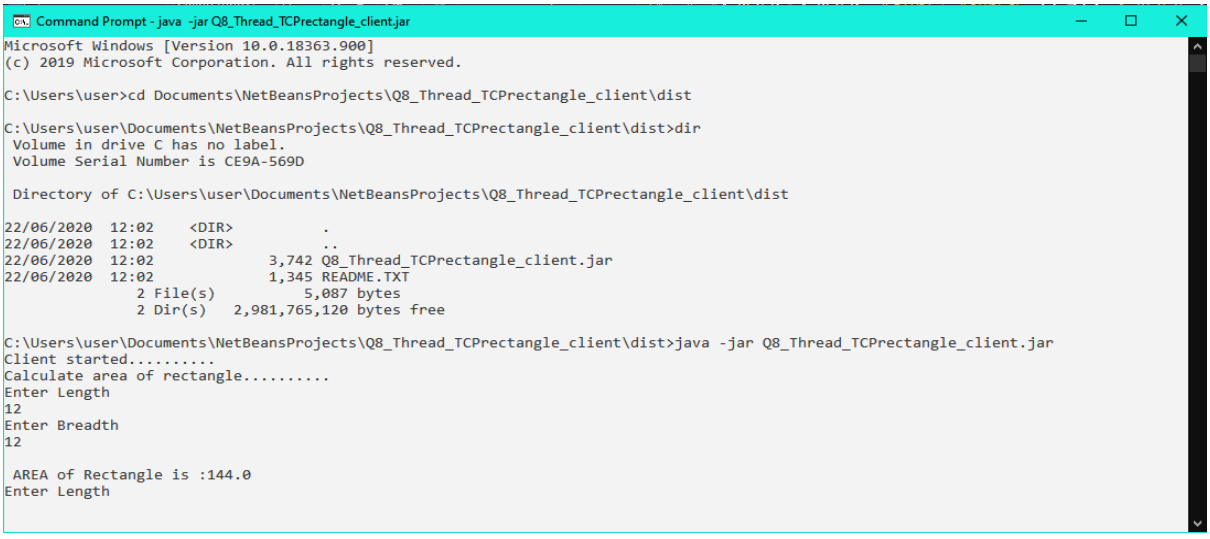
```
Scanner s = new Scanner(System.in);) {
while (true) {
System.out.println("Enter Length");
String length = s.nextLine();
out.println(length);
System.out.println("Enter Breadth");
String breadth = s.nextLine();
out.println(breadth);
if (length.equalsIgnoreCase("exit")) {
break;
}
System.out.println("\n AREA of Rectangle is :" + in.nextLine());
}
}
}

}
```

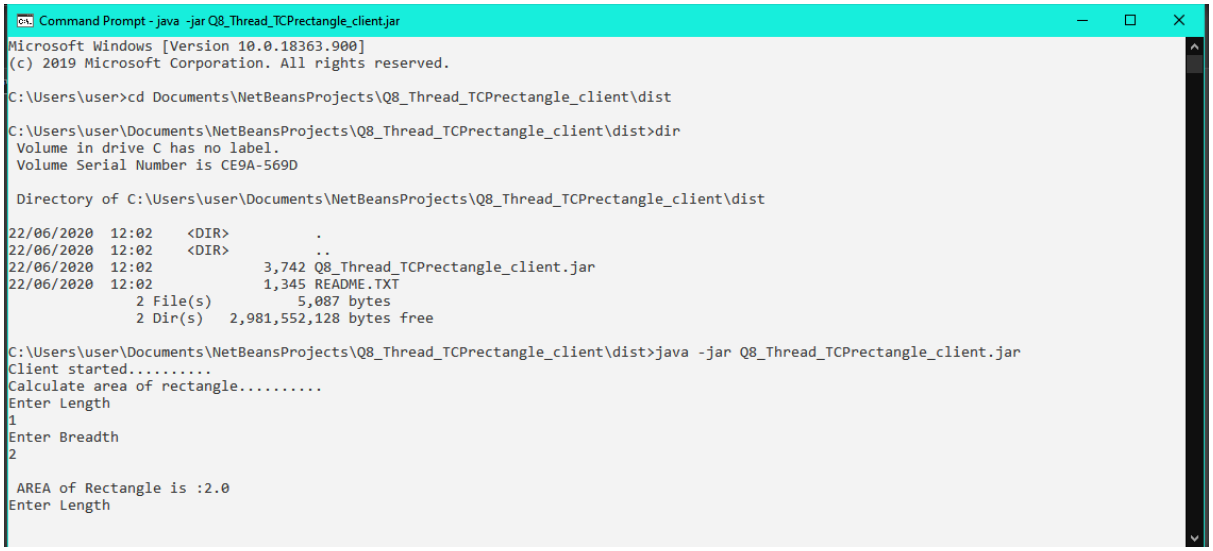
Thread_TCPrectangle_server.java

```
package thread_tcprectangle_server;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

public class Thread_TCPrectangle_server {

public static void main(String[] args) throws IOException,
NumberFormatException {
final int PORT = 1234;
ServerSocket serverSocket = new ServerSocket(PORT);

System.out.println("Server started.");
System.out.println("Listening to client.........");

while (true) {
Socket clientSocket = serverSocket.accept();
Thread t = new Thread() {
@Override
public void run() {
try (
PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
Scanner in = new Scanner(clientSocket.getInputStream());) {
while (in.hasNextLine()) {
String line1 = in.nextLine();
int length = Integer.valueOf(line1);
String line2 = in.nextLine();
```

```
int breadth = Integer.valueOf(line2);

if (line1.equalsIgnoreCase("exit")) {

break;
}

System.out.println("\nOk..I will calculate the area of rectangle for
"
+ "length =" + length + "breadth=" + breadth);

double area = length * breadth;
String inputline = Double.toString(area);

out.println(inputline);
}
} catch (IOException e) {
}
}
};
t.start();
}
}
}
```






Command Prompt - java -jar Q8_Thread_TCPrectangle_client.jar
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\user>cd Documents\NetBeansProjects\Q8_Thread_TCPrectangle_client\dist
C:\Users\user\Documents\NetBeansProjects\Q8_Thread_TCPrectangle_client\dist>dir
Volume in drive C has no label.
Volume Serial Number is CE9A-569D
Directory of C:\Users\user\Documents\NetBeansProjects\Q8_Thread_TCPrectangle_client\dist
22/06/2020 12:02 <DIR> .
22/06/2020 12:02 <DIR> ..
22/06/2020 12:02 3,742 Q8_Thread_TCPrectangle_client.jar
22/06/2020 12:02 1,345 README.TXT
2 File(s) 5,087 bytes
2 Dir(s) 2,981,552,128 bytes free
C:\Users\user\Documents\NetBeansProjects\Q8_Thread_TCPrectangle_client\dist>java -jar Q8_Thread_TCPrectangle_client.jar
Client started.....
Calculate area of rectangle.....
Enter Length
1
Enter Breadth
2
AREA of Rectangle is :2.0
Enter Length