TRINITY INTERNATIONAL COLLEGE

(Tribhuvan University Affiliated)

Lab Assignment 5: Advance Java Programming

Submitted By:

Name:

Program: **B. Sc. (CSIT)**

Roll No:

Semester: seventh (7^{th})

Date:

Submitted to:

_____

KATHMANDU, NEPAL

2020

Unit- 5 Network Programming

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

Program

Client Side

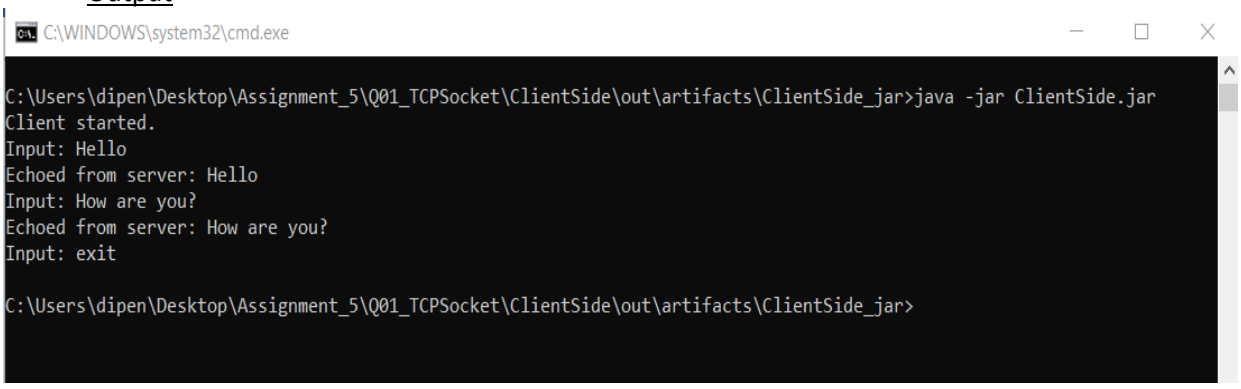
```
import java.util.Scanner;

public class 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.");
        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("Input: ");
                String input = s.nextLine();
                out.println(input);
                if (input.equalsIgnoreCase("exit")) break;

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

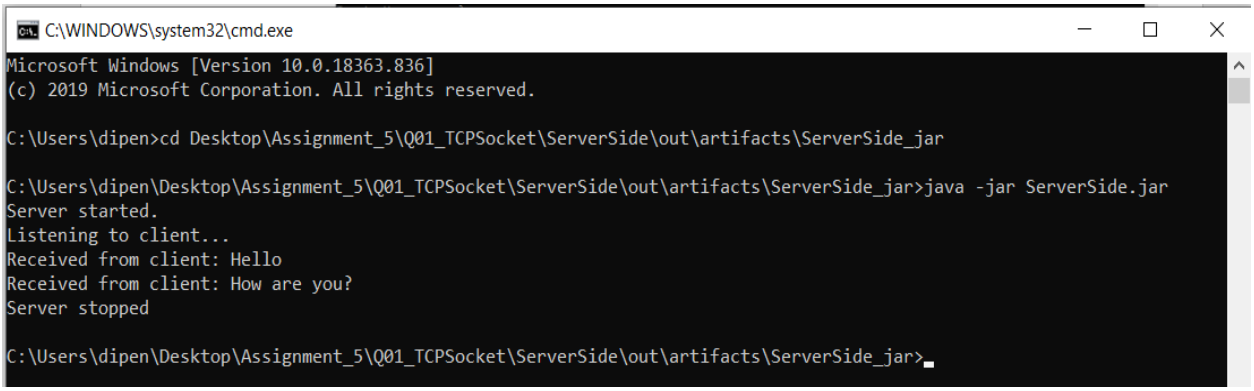
Server Side

```
package EchoServer;

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

public class Server {
    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 input = in.nextLine();
                if (input.equalsIgnoreCase("exit")) break;
                System.out.println("Received from client: "
                                                    + input);
                out.println(input);
             }
        }
        System.out.println("Server stopped");
    }
}
```

Output





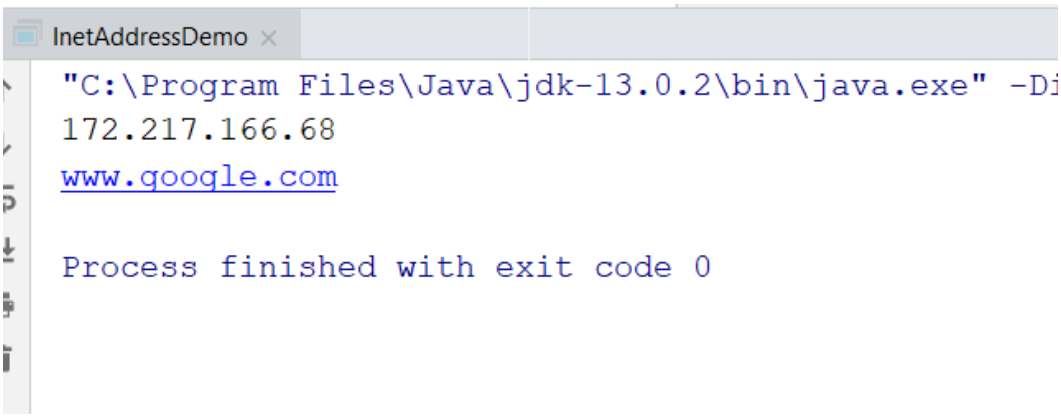
2) Write a program to illustrate the use of InetAddress class. [2073]

⇨

Program

```
package InetAddress;

import java.net.InetAddress;
import java.net.UnknownHostException;

public class InetAddressDemo
{
        public static void main(String[] args)throws
                                            UnknownHostException
        {
            InetAddress address = InetAddress.
                                    getByName("www.google.com");
            System.out.println(address.getHostAddress());
            System.out.println(address.getHostName());
        }
}
```

output



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]

⇨

Program

Client Side

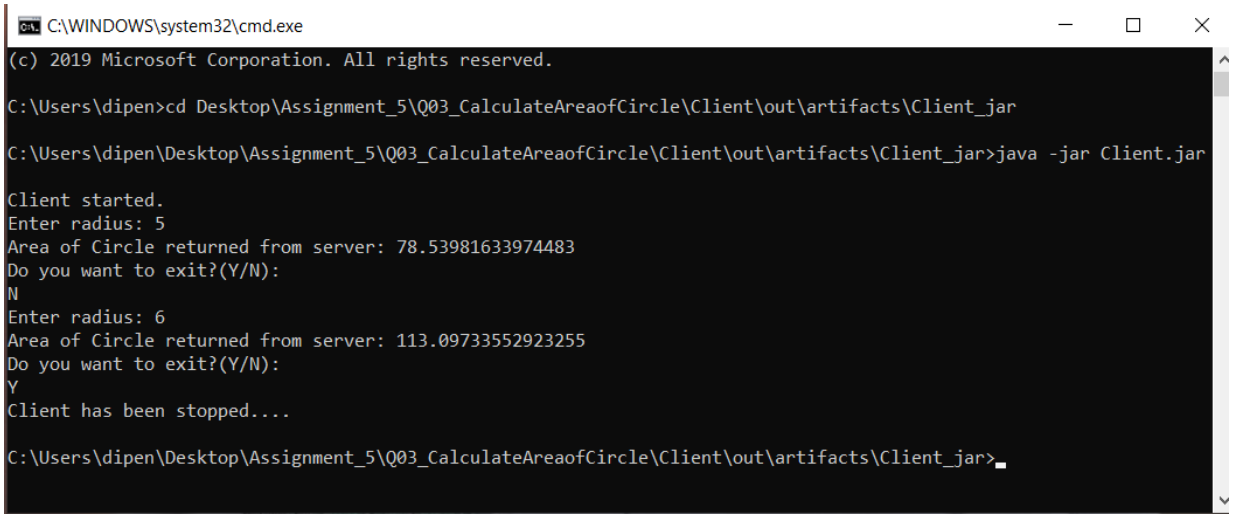
```
package AreaofCircle;

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

public class 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.");
        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("Enter radius: ");
                double radius = s.nextDouble();
                out.println(radius);
                System.out.println("Area of Circle
                                returned from server: "
                                     + in.nextDouble());
                System.out.println("Do you want to
                                            exit?(Y/N):");
                String choice = s.next();
                out.println(choice);
                if(choice.equalsIgnoreCase("Y"))
                    break;
            }
          }
       System.out.println("Client has been stopped....");
    }
}
```

Server Side

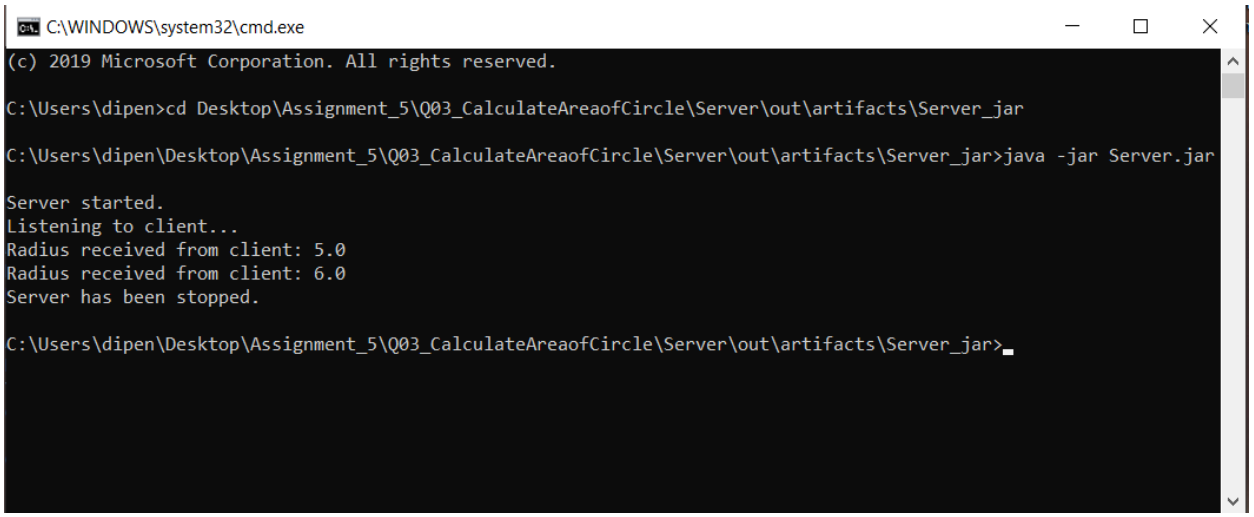
```
package AreaofCircle;

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

public class Server
 {
    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)
             {
                 double radius = in.nextDouble();
                 double area = Math.PI*radius*radius;
                 System.out.println("Radius received from
                                        client: " + radius);
                 out.println(area);

                 String choice = in.next();
                 if (choice.equalsIgnoreCase("Y")) break;
             }
            System.out.println("Server has been stopped.");
         }
     }
}
```

Output

Client Side



Server Side



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

⇨

Program

```
package EmailSMTP;

import java.io.IOException;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.util.*;
public class SendEmail
{
        public static void main(String[] args) throws
IOException          {
            Email email = new Email(
                    "dipen.stha8786@gmail.com",
                    "dipen.stha9860@hotmail.com",
                    "Test email."              );
            email.send();
        }
    }
    class Email     {
        private Scanner in = null;
        private PrintWriter out = null;
        private final String SMTP_SERVER =
                                        "smtp.wlink.com.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();
        }
}
```

Output

```
SendEmail
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe"
 220 mx-01.wlink.com.np ESMTP
>> HELO Dipendra
 250 titan-02.wlink.com.np
>> MAIL FROM: <dipen.stha8786@gmail.com>
 250 2.1.0 Ok
>> RCPT TO: <dipen.stha9860@hotmail.com>
 250 2.1.5 Ok
>> DATA
 354 End data with <CR><LF>.<CR><LF>
>> Test email.
>> .
 250 2.0.0 Ok: queued as 775C31125A2

Process finished with exit code 0
```

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.

⇨

Program

Client Side

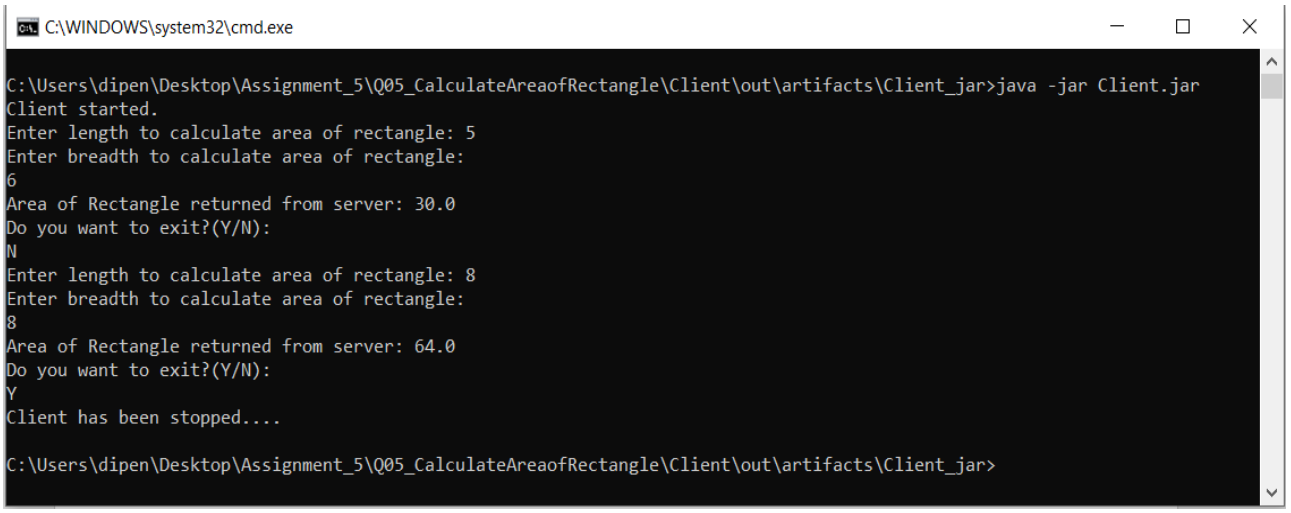
```
package AreaofRectangle;

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

public class 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.");
        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("Enter length to calculate
                                    area of rectangle: ");
                double length = s.nextDouble();
                System.out.println("Enter breadth to
                            calculate area of rectangle:");
                double breadth = s.nextDouble();
                out.println(length);
                out.println(breadth);
                System.out.println("Area of Rectangle
                                returned from server: " +
                                    in.nextDouble());
                System.out.println("Do you want to
                                            exit?(Y/N):");
                String choice = s.next();
                out.println(choice);
                if(choice.equalsIgnoreCase("Y"))
                    break;
            }
        }
        System.out.println("Client has been stopped....");
    }
}
```

Server Side

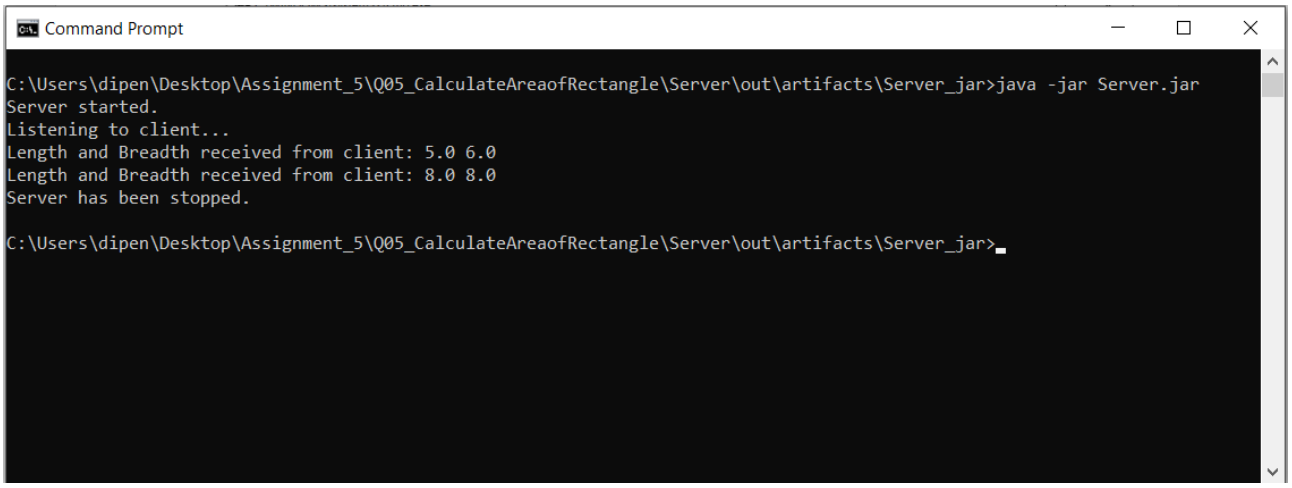
```
package AreaofRectangle;

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

public class Server
{
    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)
            {
                double length = in.nextDouble();
                double breadth = in.nextDouble();
                double area = length*breadth;

                System.out.println("Length and Breadth
                              received from client: " +
                                 length +" " + breadth);
                out.println(area);

                String choice = in.next();
                if (choice.equalsIgnoreCase("Y")) break;
            }
          System.out.println("Server has been stopped.");
        }

    }
}
```

Output

Client Side



Server Side



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

⇨

Program

Client Side

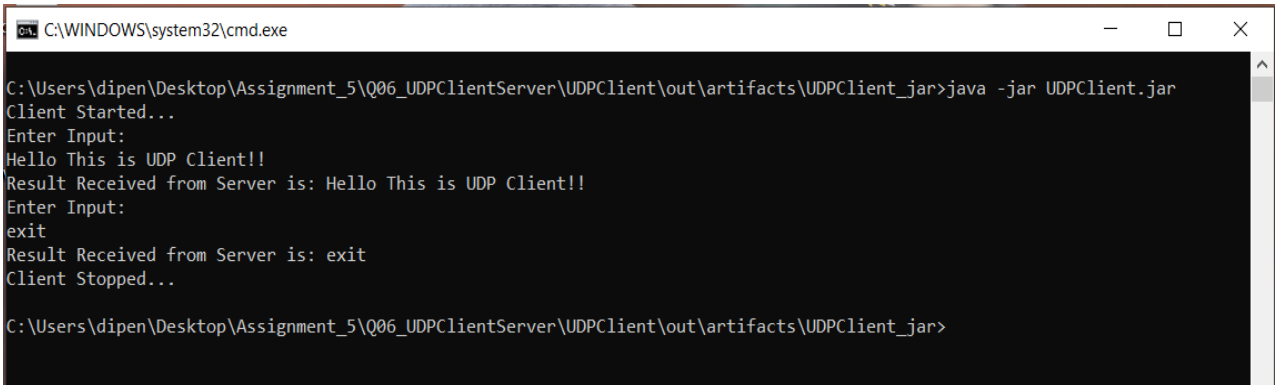
```
package Client;
import java.net.*;
import java.io.*;
public class UDPEchoClient
{
    public static void main(String[] args) throws Exception
     {
        InetAddress address = null;
        int port = 8000;
        DatagramSocket datagramSocket = null;
        BufferedReader keyboardReader = null;
       try
        {
            address = InetAddress.getByName("127.0.0.1");
            datagramSocket = new DatagramSocket();
            keyboardReader = new BufferedReader(new
                                InputStreamReader(System.in));

            System.out.println("Client Started...");
            String input;
            while (true)
            {
                System.out.println("Enter Input: ");
                input = keyboardReader.readLine();


                DatagramPacket sendDatagramPacket = new
                                DatagramPacket(input.getBytes(),
                                input.length(), address, port);
                datagramSocket.send(sendDatagramPacket);

                byte[] b1 = new byte[1024];
                DatagramPacket receiveDatagramPacket = new
                                 DatagramPacket(input.getBytes()
                                               ,input.length());
                datagramSocket.receive(receiveDatagramPacket);

                String str = new String
                               (receiveDatagramPacket.getData());
                System.out.println("Result Received from Server
                                                  is: " + str);
                if (input.equalsIgnoreCase("exit")) break;
            }
            System.out.println("Client Stopped...");
        }
        catch (IOException e)
        {
            System.out.println(e);
            System.exit(1);
        }
    }
}
```

Server Side

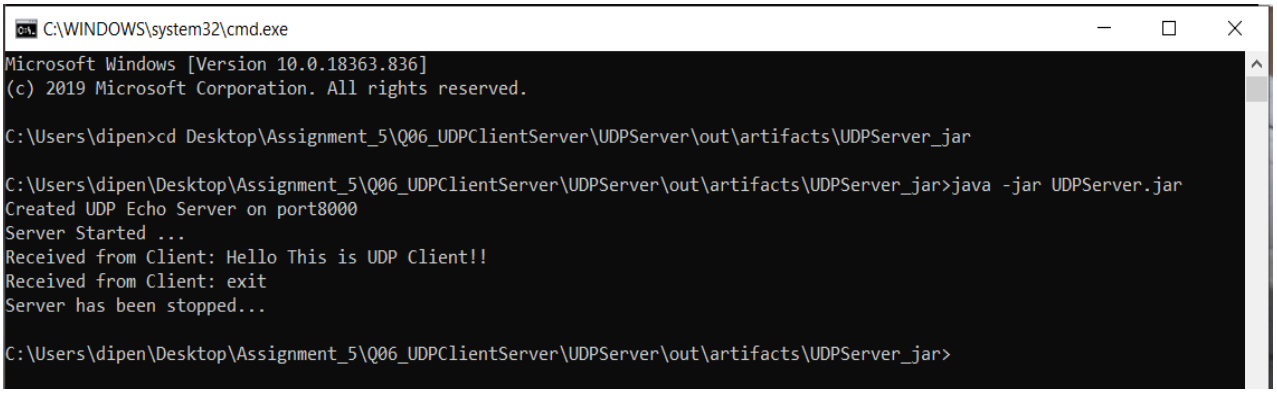
```
package Server;

import java.net.*;
import java.io.*;
public class UDPEchoServer
{
    public static void main(String args[])
    {
        int port = 8000;

        DatagramSocket serverDatagramSocket = null;

        try
        {
            serverDatagramSocket = new DatagramSocket(port);
            System.out.println("Created UDP Echo Server on
                                                    port"+port);

            byte buffer[] = new byte[1024];
            DatagramPacket datagramPacket = new DatagramPacket
                                        (buffer, buffer.length);
            System.out.println("Server Started ...");
            String input;

            while(true)
            {
                serverDatagramSocket.receive(datagramPacket);
                input = new String(datagramPacket.getData(),
                                0,datagramPacket.getLength());

                System.out.println("Received from Client: "+
                                                        input);
                serverDatagramSocket.send(datagramPacket);

                if (input.equalsIgnoreCase("exit")) break;
            }
            System.out.println("Server has been stopped...");
        }
        catch(IOException e)
        {
            System.out.println(e);
            System.exit(1);
        }
    }
}
```

Output

Client Side



Server Side



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.

⇨

Program

Client side

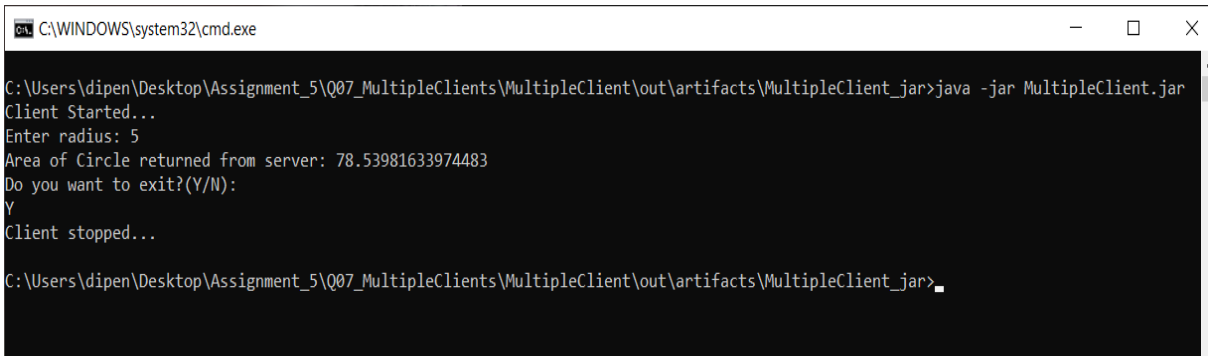
```
package MultipleClient;

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

public class ClientSide
{
    public static void main(String[] args) throws
                                                Exception
     {
        final String HOST = "localhost";
        final int PORT = 1234;

        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)
        ) {
            System.out.println("Client Started...");
            while (true)
            {
                System.out.print("Enter radius: ");
                double radius = s.nextDouble();
                out.println(radius);
                System.out.println("Area of Circle
                             returned from server: " +
                                      in.nextDouble());
            System.out.println("Do you want to
                                    exit?(Y/N): ");
                String choice = s.next();
                out.println(choice);
                if(choice.equalsIgnoreCase("Y")) break;
            }
            System.out.println("Client stopped...");

        }
        catch (Exception e)
        {

        }
    }
}
```

Server Side

```
public class Server
{
    public static void main(String[] args) throws
                                                Exception
    {
        final int PORT = 1234;
        ServerSocket serverSocket = new
                                        ServerSocket(PORT);
        System.out.println("Server Started...");
        while(true)
        {
            Socket clientSocket = serverSocket.accept();
            Thread t = new Thread()
            {
                public void run()
                {
                    try (
                            PrintWriter out = new
                                            PrintWriter
                        (clientSocket.getOutputStream(),
                                                true);
                            Scanner in = new Scanner
                                        (clientSocket.
                                    getInputStream());
                    )
                    {
                        while (true)
                        {
                            double radius =
                                        in.nextDouble();
                            double area =
                                  Math.PI*radius*radius;
                            System.out.println("Radius
                                received from client: "
                                            + radius);
                                    out.println(area);

                             String choice = in.next();
                         if(choice.equalsIgnoreCase("Y"))
                                break;
                        }
                    }
                    catch (IOException e)
                    {
                    }
                }
            };
            t.start();
        }
    }

}
```

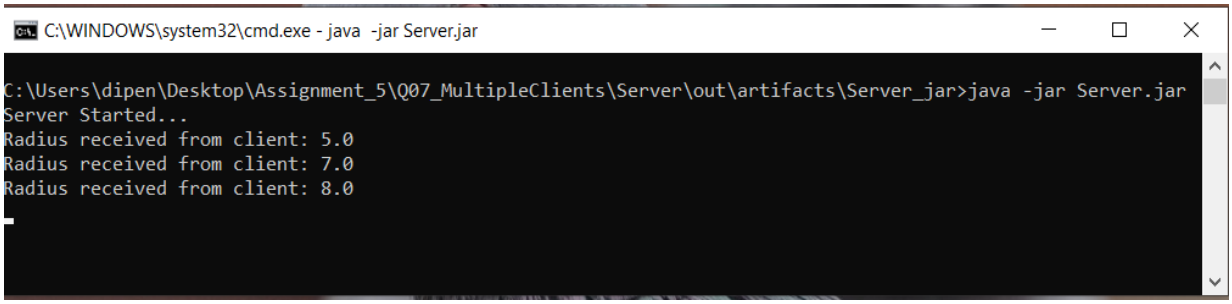
Output

1st Client



2nd Client

Server



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.
⇨

Program

Client Side

```
package MultipleClient;

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

public class Client
{
    public static void main(String[] args) throws Exception
{
        final String HOST = "localhost";
        final int PORT = 1234;
        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)
        ) {
            System.out.println("Client Started...");
            while (true)
            {
                System.out.print("Enter Length: ");
                double length = s.nextDouble();
                out.println(length);
                System.out.print("Enter Breadth: ");
                double breadth = s.nextDouble();
                out.println(breadth);
                System.out.println("Area of rectangle
                                returned from server: " +
                                        in.nextDouble());
                System.out.println("Do you want to
                                        exit?(Y/N): ");
                String choice = s.next();
                out.println(choice);
                if(choice.equalsIgnoreCase("Y")) break;
            }
            System.out.println("Client stopped...");
        }
        catch (Exception e)
        {

        }
    }
}
```

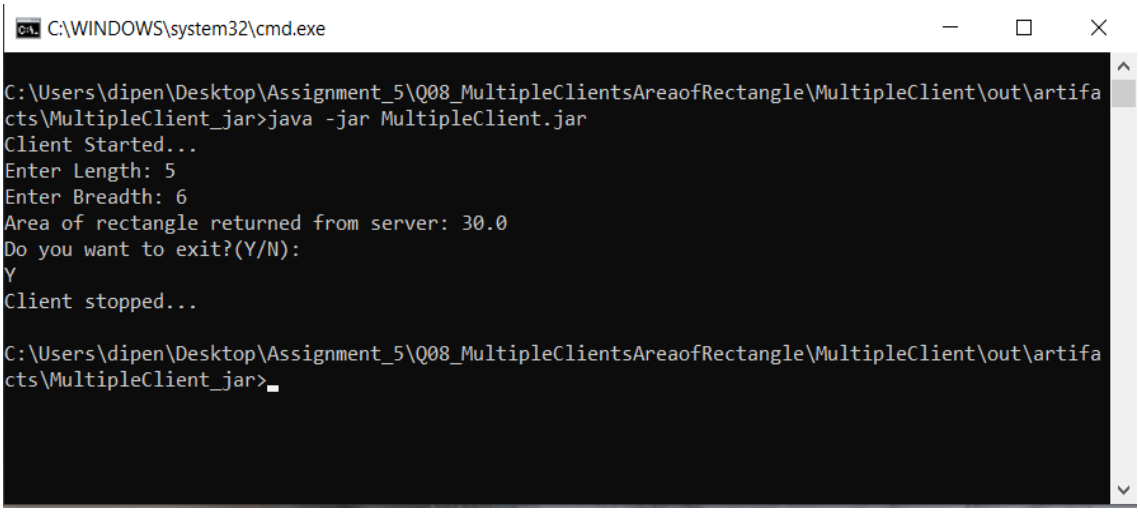
Server Side

```
package ServerSide;

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

public class Server
{
    public static void main(String[] args) throws
                                                Exception
     {
        final int PORT = 1234;
        ServerSocket serverSocket = new ServerSocket
                                                (PORT);
        System.out.println("Server Started...");
        while(true)
        {
            Socket clientSocket = serverSocket.accept();
            Thread t = new Thread()
            {
                public void run()
                {
                    try (

                            PrintWriter out = new
                               PrintWrite(clientSocket.
                                       getOutputStream(),
                                                true);
                            Scanner in = new Scanner
                                            (clientSocket.
                                     getInputStream());
                    )
                    {
                        while (true)
                        {
                            double length =
                                            in.nextDouble();
                            double breadth =
                                            in.nextDouble();
                            double area = length*breadth;
                            System.out.println("Length and
                                    Breadth received from
                                client: " + length + " " +
                                                breadth);
                            out.println(area);

                            String choice = in.next();
                        if (choice.equalsIgnoreCase("Y"))
                                break;
                        }
                    }
                    catch (IOException e)
```

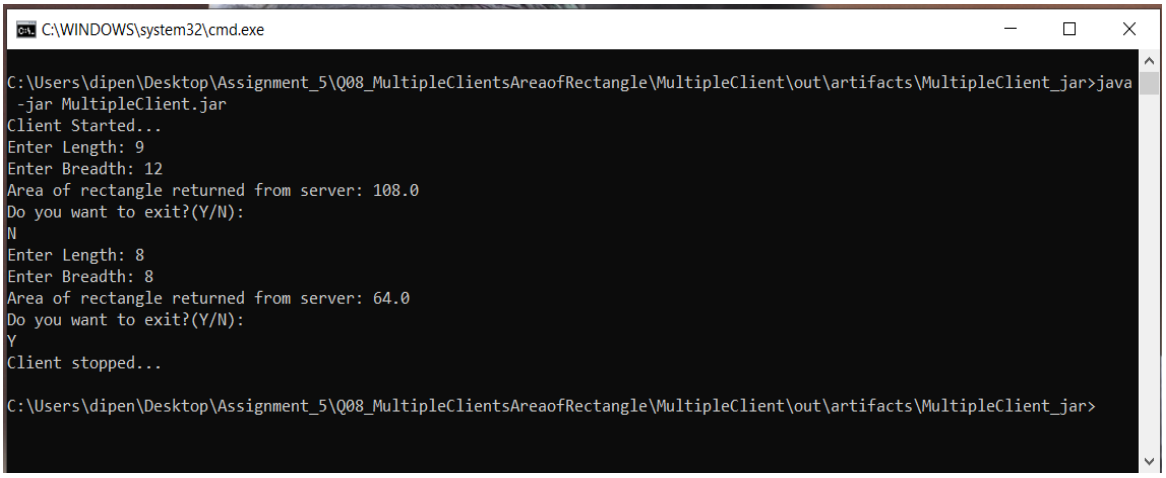
```
                        {
                        }
                    }
                };
                t.start();
            }
        }
```

Output

1st Client



2nd Client



Server