

Name : Aakash Pavar | Roll No : 3057 | DCN Assignment

Q1. Client.java

```
import java.net.*;
import java.io.*;

/**
 * Q1
 */
public class Q1Client {
    public static void main(String[] args) throws Exception {
        Socket s = new Socket("localhost", 3333);
        DataInputStream din = new DataInputStream(s.getInputStream());
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter the Number :");
        String str = br.readLine();
        dout.writeUTF(str);
        dout.flush();
        String str2 = din.readUTF();
        System.out.println("Server says Square of : " + str + " is " + str2);

        dout.close();
        s.close();
    }
}
```

Q1 Server.java

```
import java.net.*;
import java.io.*;

public class Q1Server {
    public static void main(String[] args) throws Exception {
        ServerSocket ss = new ServerSocket(3333);
        Socket s = ss.accept();
        DataInputStream din = new DataInputStream(s.getInputStream());
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        // BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
    }
}
```

```

        String str = din.readUTF();
        System.out.println("client says: " + str);
        int ans = Integer.parseInt(str);
        String str2 = Integer.toString(ans*ans);
        dout.writeUTF(str2);
        dout.flush();

        din.close();
        s.close();
        ss.close();
    }
}

```

Q2 Client.java

```

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

public class Q2Client {
    public static void main(String[] args) {
        try{
            DatagramSocket ss = new DatagramSocket();
            InetAddress serverAddress = InetAddress.getByName("localhost");
            int serverport=9876;

            Scanner scan = new Scanner(System.in);
            System.out.println("Enter The Number : ");
            String msg = scan.nextLine();

            byte[] sendData = msg.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, serverport);
            ss.send(sendPacket);

            byte[] receiveData = new byte[4000];
            DatagramPacket recieverPacket = new DatagramPacket(receiveData,
receiveData.length);
            ss.receive(recieverPacket);

```

```

        String receivedMsg = new
String(recieverPacket.getData(),0,recieverPacket.getLength());
        // String receivedMsg = new String(receiveData, 0,
receiveData.length);
        System.out.println("Server Says Cube of " + msg + " is " +
receivedMsg);

        ss.close();
        scan.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
}

```

Q2 Server.java

```

import java.net.*;

public class Q2Server{
    public static void main(String[] args) {
        DatagramSocket socket = null;

        try {

            socket = new DatagramSocket(9876);

            byte[] receiveData = new byte[1024];

            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

            System.out.println("Server started...");

            while (true) {
                socket.receive(receivePacket);
            }
        }
    }
}

```

```

        String receivedMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());

        int ans = Integer.parseInt(receivedMessage);
        System.out.println("Recieved From Client :"+receivedMessage);
        String msg1=Integer.toString( ans*ans*ans);

        byte[] sendData = msg1.getBytes();

        InetAddress clientAddress = receivePacket.getAddress();
        int clientPort = receivePacket.getPort();

        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);

        socket.send(sendPacket);
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    finally {
        if (socket != null) {
            socket.close();
        }
    }
}
}

```

Q3Client.java

```

import java.net.*;
import java.util.Scanner;
import java.io.*;

public class Q3Client {
    public static void main(String[] args) throws Exception {
        Socket ss = new Socket("localhost", 1234);
        PrintWriter out = new PrintWriter(ss.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
InputStreamReader(ss.getInputStream()));
    }
}

```

```

Scanner scan = new Scanner(System.in);
try {
    System.out.println("Enter a set of numeric values separated by comma:");

    String msg = scan.nextLine();
    out.println(msg);

    String primeNumbers = in.readLine();
    if (primeNumbers.equals("0")) {
        System.out.println("No prime numbers found in the set.");
    } else {
        System.out.println("Prime numbers from the set: " +
primeNumbers);
    }

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        scan.close();
        in.close();
        out.close();
        ss.close();
    }
}
}

```

Q3 Server.java

```

import java.net.*;
import java.util.ArrayList;
import java.util.Scanner;
import java.io.*;

public class Q3Server {
    public static boolean isPrime(int num) {
        if (num <= 1)
            return false;
        if (num == 2)
            return true;
        if (num % 2 == 0)
            return false;
        for (int i = 3; i <= Math.sqrt(num); i += 2) {

```

```

        if (num % i == 0)
            return false;
    }
    return true;
}

public static String getPrimeNumbers(String data) {
    String[] numbersStr = data.split(",");
    ArrayList<Integer> numbers = new ArrayList<>();
    for (String numStr : numbersStr) {
        numbers.add(Integer.parseInt(numStr));
    }

    ArrayList<Integer> primes = new ArrayList<>();
    for (int num : numbers) {
        if (isPrime(num)) {
            primes.add(num);
        }
    }

    if (primes.isEmpty()) {
        return "0";
    } else {
        StringBuilder result = new StringBuilder();
        for (int prime : primes) {
            result.append(prime).append(",");
        }
        return result.substring(0, result.length() - 1);
    }
}

public static void main(String[] args) throws IOException {
    ServerSocket ss = new ServerSocket(1234);
    Socket cs = ss.accept();
    System.out.println("client is connected");
    PrintWriter out = new PrintWriter(cs.getOutputStream(), true);
    BufferedReader in = new BufferedReader(new
InputStreamReader(cs.getInputStream()));
    Scanner scan = new Scanner(System.in);
    try {
        String msg = in.readLine();
        System.out.println("Received data from client: " + msg);

        String primeNumbers = getPrimeNumbers(msg);
        out.println(primeNumbers);
    }
}

```

```

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            scan.close();
            in.close();
            out.close();
            ss.close();
        }
    }
}

```

Q4 Client.java

```

import java.io.*;
import java.net.*;

public class Q4Client {
    public static void main(String[] args) {
        try {
            BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
            System.out.print("Enter a set of numeric values separated by comma:
");

            String numbers = userInput.readLine();

            DatagramSocket clientSocket = new DatagramSocket();
            InetAddress serverAddress = InetAddress.getByName("localhost");
            byte[] sendData = numbers.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, 9999);
            clientSocket.send(sendPacket);

            byte[] receiveData = new byte[1024];
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            clientSocket.receive(receivePacket);

            String quartilesStr = new String(receivePacket.getData(), 0,
receivePacket.getLength());

```

```

        System.out.println("Quartiles received from server: " +
quartilesStr);

        clientSocket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Q4 Server.java

```

import java.io.*;
import java.net.*;
import java.util.*;

public class Q4Server {
    public static List<Integer> calculateQuartiles(List<Integer> data) {
        Collections.sort(data);
        int n = data.size();
        int q1Index = n / 4;
        int q2Index = n / 2;
        int q3Index = (3 * n) / 4;

        int q1 = data.get(q1Index);
        int q2 = data.get(q2Index);
        int q3 = data.get(q3Index);

        List<Integer> quartiles = new ArrayList<>();
        quartiles.add(q1);
        quartiles.add(q2);
        quartiles.add(q3);
        return quartiles;
    }

    public static void main(String[] args) {
        try {
            DatagramSocket serverSocket = new DatagramSocket(9999);
            byte[] receiveData = new byte[1024];

            while (true) {
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

```



```

        serverSocket.receive(receivePacket);

        String inputData = new String(receivePacket.getData(), 0,
receivePacket.getLength());
        System.out.println("Received data from client: " + inputData);

        String[] numbersStr = inputData.split(",");
        List<Integer> numbers = new ArrayList<>();
        for (String numStr : numbersStr) {
            numbers.add(Integer.parseInt(numStr));
        }

        List<Integer> quartiles = calculateQuartiles(numbers);
        String quartilesStr = String.join(",",
quartiles.stream().map(Object::toString).toArray(String[]::new));

        InetAddress clientAddress = receivePacket.getAddress();
        int clientPort = receivePacket.getPort();
        byte[] sendData = quartilesStr.getBytes();
        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
        serverSocket.send(sendPacket);
    }
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```

Q5 Client.java

```

import java.io.*;
import java.net.*;

public class Q5Client {
    public static void main(String[] args) throws SocketException {

        try {
            DatagramSocket ds = new DatagramSocket();
            InetAddress ina = InetAddress.getByName("localhost");
            byte[] sendData = new byte[1024];

```

```

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

        while (true) {
            System.out.println("Options:");
            System.out.println("a. Celsius to Fahrenheit");
            System.out.println("b. Fahrenheit to Celsius");
            System.out.println("c. Exit");
            System.out.print("Enter your choice: ");

            String option = userInput.readLine();

            if (option.equals("c")) {
                break;
            }

            System.out.print("Enter temperature: ");
            double temperature = Double.parseDouble(userInput.readLine());

            String inputData = option + ":" + temperature;
            sendData = inputData.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, ina, 9999);
            ds.send(sendPacket);

            byte[] receiveData = new byte[1024];
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            ds.receive(receivePacket);

            String response = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("Converted temperature: " + response);
        }
        ds.close();

    } catch (Exception e) {
        e.printStackTrace();
    }

}
}

```

Q5 Server.java

```
import java.net.*;

public class Q5Server {
    public static double celsiusToFahrenheit(double celsius) {
        return (celsius * 9 / 5) + 32;
    }

    public static double fahrenheitToCelsius(double fahrenheit) {
        return (fahrenheit - 32) * 5 / 9;
    }

    public static void main(String[] args) {
        try {
            DatagramSocket serverSocket = new DatagramSocket(9999);
            byte[] receiveData = new byte[1024];
            System.out.println("Server listening on port 9999...");

            while (true) {
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
                    receiveData.length);
                serverSocket.receive(receivePacket);

                String msg = new String(receivePacket.getData(), 0,
                    receivePacket.getLength());
                System.out.println("Received Data From Client : " + msg);

                String[] inputParts = msg.split(":");
                String option = inputParts[0];
                double temperature = Double.parseDouble(inputParts[1]);
                double convertedTemperature = 0;

                switch (option) {
                    case "a":
                        convertedTemperature = celsiusToFahrenheit(temperature);
                        break;
                    case "b":
                        convertedTemperature = fahrenheitToCelsius(temperature);
                        break;
                }
                String response = Double.toString(convertedTemperature);
                InetAddress clientAddress = receivePacket.getAddress();
                int clientPort = receivePacket.getPort();
                byte[] sendData = response.getBytes();
```

```

        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);
        serverSocket.send(sendPacket);
    }

}
catch(Exception e)
{
    e.printStackTrace();
}
}
}

```

Q6 Client.java

```

import java.io.*;
import java.net.*;

public class Q6Client {
    public static void main(String[] args) {
        try{
            Socket ss = new Socket("localhost",1234);
            BufferedReader in = new BufferedReader(new
InputStreamReader(ss.getInputStream()));
            PrintWriter out = new PrintWriter(ss.getOutputStream(),true);
            BufferedReader inputData = new BufferedReader(new
InputStreamReader(System.in));
            while (true) {
                System.out.println("Options:");
                System.out.println("a. Circle");
                System.out.println("b. Rectangle");
                System.out.println("c. Square");
                System.out.println("d. Triangle");
                System.out.println("e. exit");
                System.out.print("Enter your choice: ");

                String option = inputData.readLine();

                if (option.equals("e")) {
                    out.println(option);
                    break;
                }
            }
        }
    }
}

```

```

        if (option.equals("a")) {
            System.out.print("Enter Radius For Circle :");
            String r = inputData.readLine();
            String sendData = option+": "+r;
            out.println(sendData);
        }
        else if (option.equals("b")) {
            System.out.print("Enter Value Of height For Rectangle :");
            String l = inputData.readLine();
            System.out.print("Enter Value Of Width For Rectangle :");
            String w = inputData.readLine();
            String sendData = option+": "+l+": "+w;
            out.println(sendData);
        }
        else if (option.equals("c")) {
            System.out.print("Enter Value Of Side For Square :");
            String s = inputData.readLine();
            String sendData = option+": "+s;
            out.println(sendData);
        }
        else if (option.equals("d")) {
            System.out.print("Enter Value Of Base For Triangle :");
            String b = inputData.readLine();
            System.out.print("Enter Value Of height For Triangle :");
            String h = inputData.readLine();
            String sendData = option+": "+b+": "+h;
            out.println(sendData);
        }

        String response = in.readLine();
        System.out.println("Server Says : "+response);

    }
}
catch(Exception e)
{
}
}
}

```

Q6 Server.java

```
import java.io.*;
import java.net.*;
// import java.math.*;

public class Q6Server {
    public static double area(double s)
    {
        return s*s;
    }
    public static double areas(double r)
    {
        return Math.PI*r*r;
    }
    public static double area(double w,double l)
    {
        return w*l;
    }
    public static double area(double p,double b,double h)
    {
        return p*b*h;
    }
    public static void main(String[] args) {
        try{
            ServerSocket ss = new ServerSocket(1234);
            Socket cs = ss.accept();
            BufferedReader in = new BufferedReader( new
InputStreamReader(cs.getInputStream()));
            PrintWriter out = new PrintWriter(cs.getOutputStream(),true);
            System.out.println("Server is Listening....");
            while (true) {

                String receiveData = in.readLine();
                System.out.println("Client Says :"+receiveData);
                String[] data = receiveData.split(":");
                if (receiveData.equals("e")) {
                    break;
                }
                if (data[0].equals("a")) {
                    double r= Double.parseDouble(data[1]);
                    double response = areas(r);
                    String sendData = "Server Says Area of Circle is :"+response;
                    out.println(sendData);
                }
            }
        }
    }
}
```

```

        else if (data[0].equals("b")) {
            double l= Double.parseDouble(data[1]);
            double w= Double.parseDouble(data[2]);
            double response = area(l,w);
            String sendData = "Server Says Area of Rectangle is
:"+response;
            out.println(sendData);
        }
        else if (data[0].equals("c")) {
            double r= Double.parseDouble(data[1]);
            double response = area(r);
            String sendData = "Server Says Area of Square is :"+response;
            out.println(sendData);
        }
        else if (data[0].equals("d")) {
            double b= Double.parseDouble(data[1]);
            double h= Double.parseDouble(data[2]);
            double response = area(0.5,b,h);
            String sendData = "Server Says Area of Rectangle is
:"+response;
            out.println(sendData);
        }
    }
}
catch(Exception e)
{
    e.printStackTrace();
}
}
}

```

Q7 Client.java

```

import java.io.*;
import java.net.*;

public class Q7Client {
    public static void main(String[] args) {
        try {
            BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));

```

```

        System.out.print("Enter a sentence: ");
        String sentence = userInput.readLine();

        Socket socket = new Socket("localhost", 9999);
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

        out.println(sentence);

        String camelCase = in.readLine();
        System.out.println("Camel case response from server: " + camelCase);

        socket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Q7 Server.java

```

import java.io.*;
import java.net.*;

public class Q7Server {
    public static String toCamelCase(String sentence) {
        StringBuilder camelCase = new StringBuilder();
        String[] words = sentence.split("\\s+");
        for (int i = 0; i < words.length; i++) {
            // if (i == 0) {
            //     camelCase.append(words[i].toLowerCase());
            // } else {
            //     camelCase.append(words[i].substring(0, 1).toUpperCase())
            //         .append(words[i].substring(1).toLowerCase()).append(" ");
            // }
            camelCase.append(words[i].substring(0, 1).toUpperCase())
                .append(words[i].substring(1).toLowerCase()).append(" ");
        }
        return camelCase.toString();
    }
}

```



```

    public static void main(String[] args) throws Exception{
        ServerSocket serverSocket = new ServerSocket(9999);
        System.out.println("Server listening on port 9999...");
        try {

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("Connection from " +
clientSocket.getInetAddress().getHostAddress());

                BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
                PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);

                String sentence = in.readLine();
                System.out.println("Received sentence from client: " + sentence);

                String camelCase = toCamelCase(sentence);
                out.println(camelCase);

                clientSocket.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (serverSocket != null) {
                serverSocket.close();
            }
        }
    }
}

```

Q8 Client.java

```

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

public class Q8Client {
    public static void main(String[] args) {

```

```

    try{
        DatagramSocket ss = new DatagramSocket();
        InetAddress serverAddress = InetAddress.getByName("localhost");
        int serverport=9876;

        Scanner scan = new Scanner(System.in);
        System.out.println("Enter a sentence: ");
        String msg = scan.nextLine();

        byte[] sendData = msg.getBytes();
        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverAddress, serverport);
        ss.send(sendPacket);

        byte[] receiveData = new byte[4000];
        DatagramPacket recieverPacket = new DatagramPacket(receiveData,
receiveData.length);
        ss.receive(recieverPacket);

        String receivedMsg = new
String(recieverPacket.getData(),0,recieverPacket.getLength());
        System.out.println("Toggle case response from server: " +
receivedMsg);

        ss.close();
        scan.close();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
}

```

Q8 Server.java

```

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

public class Q8Server {
    public static String toToggleCase(String sentence) {
        StringBuilder toggleCase = new StringBuilder();
    }
}

```

```

String[] words = sentence.split("\\s+");

for (int i = 0; i < words.length; i++) {
    for (char c : words[i].toCharArray()) {
        if (Character.isUpperCase(c)) {
            toggleCase.append(Character.toLowerCase(c));
        } else if (Character.isLowerCase(c)) {
            toggleCase.append(Character.toUpperCase(c));
        } else {
            toggleCase.append(c);
        }
    }
    toggleCase.append(" ");
}
return toggleCase.toString();
}

public static void main(String[] args) {
    DatagramSocket socket = null;

    try {

        socket = new DatagramSocket(9876);

        byte[] receiveData = new byte[1024];

        DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

        System.out.println("Server started...");

        while (true) {
            socket.receive(receivePacket);

            String receivedMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());

            System.out.println("Recieved From Client : " + receivedMessage);

            String msg1 = toToggleCase(receivedMessage);

            byte[] sendData = msg1.getBytes();

            InetAddress clientAddress = receivePacket.getAddress();

```

```
        int clientPort = receivePacket.getPort();

        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, clientAddress, clientPort);

        socket.send(sendPacket);
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (socket != null) {
        socket.close();
    }
}
}
```