

Name: Chavda Mamta Maheshbhai

div:B

Roll no:3100

DCN ASSIGNMENT1

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File, Edit, Selection, View, Go, ...
- Search Bar:** DCN_assignment
- Toolbars:** Standard toolbar icons for copy, paste, search, etc.
- Project Explorer:** Shows files: Server01_TCP.java, Untitled-1, Server01_TCPclass, Client01_TCP[1].java, Client01_TCP.java, Client01_TCP.java, and another Client01_TCP.java entry.
- Code Editor:** Displays the `Server01_TCP.java` file content.
- Code Content:**

```
1 // import java.util.Scanner;
2 import java.net.*;
3 import java.io.*;
4
5 class Server01_TCP {
6     Run | Debug
7     public static void main(String args[]) throws Exception {
8         ServerSocket serverSocket = new ServerSocket(port:3176);
9         System.out.println(x:"\nServer is running and waiting for client connection...");
10        Socket socket = serverSocket.accept();
11        DataInputStream din = new DataInputStream(socket.getInputStream());
12        DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
13        BufferedReader bufferReader = new BufferedReader(new InputStreamReader(System.in));
14        // Server TCP
15        System.out.println("Connected to the Client at InetAddress:" + socket.getInetAddress() + " and Port:"
16                           + socket.getPort());
17
18        String str = "Hello";
19        while (true) {
20            str = din.readUTF();
21            if (str.equalsIgnoreCase(anotherString:"stop")) {
22                break;
23            }
24            if (verifyNumber(str)) {
```

- Status Bar:** Ln 1, Col 29, Spaces: 4, UTF-8, CRLF, {}, Java, and a refresh icon.

A screenshot of a Java code editor interface, likely Eclipse or a similar environment, displaying a file named `Server01_TCP.java`. The code implements a TCP server that reads from a socket, processes the input, and writes a modified value back to the client. The code uses `ObjectInputStream` and `ObjectOutputStream` for reading and writing objects.

```
String str = "Hello";
while (true) {
    str = din.readUTF();
    if (str.equalsIgnoreCase("stop")) {
        break;
    }
    if (varifyNumber(str)) {
        int number = Integer.parseInt(str);
        number = number * number;
        dout.writeUTF(String.valueOf(number));
    } else {
        dout.writeUTF(str:"Invalid Input");
    }
}
System.out.println(x:"End of the Program.");
din.close();
dout.close();
bufferReader.close();
serverSocket.close();
```

The code editor features a dark theme with syntax highlighting for Java keywords and comments. A vertical toolbar on the left contains icons for file operations, search, and other development tools. The status bar at the bottom shows the current line (Ln 38), column (Col 6), and encoding (UTF-8). The title bar indicates the project name is `DCN_assignment`.

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File, Edit, Selection, View, Go, ...
- Search Bar:** DCN_assignment
- Toolbar:** Includes icons for New, Open, Save, Copy, Paste, Find, Replace, Cut, Copy, Paste, Undo, Redo, Run, Debug, and Build.
- Code Editor:** The active tab is Client01_TCP.java. The code is as follows:

```
Client01_TCP.java > ...
import java.util.Scanner;
import java.net.*;
import java.io.*;
class Client01_TCP {
    public static void main(String args[]) throws Exception {
        Socket socket = new Socket("localhost", 3176);
        DataInputStream din = new DataInputStream(socket.getInputStream());
        DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
        BufferedReader bufferReader = new BufferedReader(new InputStreamReader(System.in));
        Scanner scanner = new Scanner(System.in);
        // Client TCP
        System.out.println("\nConnected to the server at InetAddress:" + socket.getInetAddress() + " and Port:"
                           + socket.getPort());
        String str = "Hello";
        while (true) {
            System.out.print("Enter Number for find a square : ");
            str = scanner.next();
            dout.writeUTF(str);
            if (str.equalsIgnoreCase("stop")) {
                break;
            }
        }
    }
}
```

The code implements a TCP client that connects to a server running on localhost port 3176. It reads user input from the console, sends it to the server via DataOutputStream, and receives responses from the server via DataInputStream. The client continues to loop until the user enters "stop".

The screenshot shows a Java code editor interface with a dark theme. The title bar reads "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, Replace, and Preferences. The top menu bar includes File, Edit, Selection, View, Go, and a Help icon. The tab bar shows several files: "Server01_TCP.java", "Untitled-1", "Server01_TCPclass", "Client01_TCP[1].java", "Client01_TCP.java", and "Client01_TCP.java". The main editor area displays the following Java code:

```
13     System.out.println("Reconnected to the server at InetAddress: " + socket.getInetAddress() + " and port: "
14             + socket.getPort());
15
16     String str = "Hello";
17     while (true) {
18         System.out.print(s:"\n\nEnter Number for find a square : ");
19         str = scanner.next();
20         dout.writeUTF(str);
21         if (str.equalsIgnoreCase(anotherString:"stop")) {
22             break;
23         }
24         dout.flush();
25         str = din.readUTF();
26         System.out.println("Square of The number is : " + str);
27     }
28
29     System.out.println("End of the Program.");
30     din.close();
31     dout.close();
32     bufferedReader.close();
33     socket.close();
34     scanner.close();
35 }
36 }
```

The status bar at the bottom indicates the current line (Ln 4, Col 1), spaces (Spaces: 4), encoding (UTF-8 CRLF), and file type (Java). There is also a small search icon.

```
C:\Windows\System32\cmd.e + X
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MAMTA\Desktop\DCN_assignment>java Client01_TCP.java

C:\Users\MAMTA\Desktop\DCN_assignment>java Client01_TCP

Connected to the server at InetAddress:localhost/127.0.0.1 and Port:3176

Enter Number for find a squire : 25
Square of The number is : 625

Enter Number for find a square : 23
Square of The number is : 529

Enter Number for find a square : 12
Square of The number is : 144

Enter Number for find a square : 11
Square of The number is : 121

Enter Number for find a square : 0
Square of The number is : 0

Enter Number for find a square : -1
Square of The number is : Invalid Input

Enter Number for find a square : n
Square of The number is : Invalid Input

Enter Number for find a square : lkoiiu
Square of The number is : Invalid Input
```

```
C:\Windows\System32\cmd.e × + | v - ⌂ X
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MAMTA\Desktop\DCN_assignment>javaC Server01_TCP.java

C:\Users\MAMTA\Desktop\DCN_assignment>java Server01_TCP

Server is running and waiting for client connection...Connected to the Client at InetAddress:/127.0.0.1 and Port:50960
End of the Program.

C:\Users\MAMTA\Desktop\DCN_assignment>
```

q2

The screenshot shows a Java code editor interface with a dark theme. The title bar reads "DCN_assignment". The left sidebar contains icons for file operations, search, and other development tools. The main area displays a Java file named "Server02_UDP.java". The code implements a UDP server that listens on port 3176. It receives data from clients, prints it to the console, and if the received string is "stop", it breaks out of the loop. If the string is a valid number, it multiplies it by itself three times.

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

class Server02_UDP {
    public static void main(String args[]) {
        try {
            DatagramSocket serverSocket = new DatagramSocket(port:3176);
            byte b[] = new byte[1024];
            // Server UDP
            System.out.println("Server is running...");

            while (true) {
                DatagramPacket receivePacket = new DatagramPacket(b, b.length);
                serverSocket.receive(receivePacket);
                String str = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
                System.out.println("Receive from Client : " + str);
                if (str.equalsIgnoreCase(anotherString:"stop")) {
                    break;
                }

                if (varifyNumber(str)) {
                    int number = Integer.parseInt(str);
                    number = number * number * number;
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

A screenshot of a Java code editor interface. The title bar reads "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The main area shows the code for "Server02_UDP.java". The code implements a UDP server that receives a number from a client, adds it to a running total, and sends the result back to the client. It includes a helper method "verifyNumber" to check if the input string contains only digits.

```
number = number + number + number;
str = String.valueOf(number);
} else {
    str = "Invalid Input";
}
b = str.getBytes();
DatagramPacket sendPacket = new DatagramPacket(b, b.length, receivePacket.getAddress(),
                                                receivePacket.getPort());
serverSocket.send(sendPacket);
}
serverSocket.close();
} catch (Exception e) {
    e.printStackTrace();
}

}

public static boolean verifyNumber(String str) {
    for (int i = 0; i < str.length(); i++) {
        if (!Character.isDigit(str.charAt(i))) {
            return false;
        }
    }
    return true;
}
```

The status bar at the bottom shows "Ln 3, Col 1" and other settings like "Spaces: 4", "UTF-8", "CRLF", and "Java".

A screenshot of a Java code editor interface. The title bar shows the project name "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The main area displays the code for "Server02_UDP.java". The code is as follows:

```
27     }
28     b = str.getBytes();
29     DatagramPacket sendPacket = new DatagramPacket(b, b.length, receivePacket.getAddress(),
30                                                     receivePacket.getPort());
31
32     serverSocket.send(sendPacket);
33 }
34 serverSocket.close();
35 } catch (Exception e) {
36     e.printStackTrace();
37 }
38 }

39
40 public static boolean verifyNumber(String str) {
41     for (int i = 0; i < str.length(); i++) {
42         if (!Character.isDigit(str.charAt(i))) {
43             return false;
44         }
45     }
46     return true;
47 }
48 }
49 }
```

The code implements a UDP server. It receives a packet, converts its bytes to a string, creates a new DatagramPacket with the same address and port, sends it back, and then closes the server socket. It also includes a static method to verify if a given string is a number by checking each character.

The screenshot shows a Java code editor interface with a dark theme. The title bar displays the project name "DCN_assignment". The left sidebar contains various icons for file operations like copy, paste, search, and run. The main workspace shows the code for "Client02_UDP.java". The code implements a UDP client to interact with a server. It uses `DatagramSocket` to send and receive data over UDP. The client connects to the local host at port 3176 and sends a message to request a cube. It also includes logic to stop the process if the server responds with "stop".

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

class Client02_UDP {
    public static void main(String args[]) {
        try {
            DatagramSocket clientSocket = new DatagramSocket();
            InetAddress address = InetAddress.getByName("localhost");
            int port = 3176;
            Scanner scanner = new Scanner(System.in);
            // Client UDP
            System.out.println("\nServer is connected at InetAddress : " + address + " and Port at : " + port);
            while (true) {
                System.out.print("Enter Number for Find Cube : ");
                String str = scanner.next();
                byte b[] = str.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(b, b.length, address, port);
                clientSocket.send(sendPacket);
                if (str.equalsIgnoreCase("stop")) {
                    break;
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

File Edit Selection View Go ... ⏪ ⏩ 🔍 DCN_assignment

最大化 最小化 最小化 关闭

Client02_UDP.java > Client02_UDP

```
17     byte b[] = str.getBytes();
18     DatagramPacket sendPacket = new DatagramPacket(b, b.length, address, port);
19     clientSocket.send(sendPacket);
20     if (str.equalsIgnoreCase(anotherString:"stop")) {
21         break;
22     }
23
24     byte b1[] = new byte[1024];
25     DatagramPacket receivePacket = new DatagramPacket(b1, b1.length);
26     clientSocket.receive(receivePacket);
27     str = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
28
29     System.out.println("Cube of Number : " + str);
30     if (str.equalsIgnoreCase(anotherString:"stop")) {
31         break;
32     }
33 }
34 scanner.close();
35 clientSocket.close();
36 } catch (Exception e) {
37     e.printStackTrace();
38 }
39 }
```

× 0 ▲ 2 ⌂ 0

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF {} Java ⚡

```
C:\Users\MAMTA\Desktop\DCN_assignment>javaC Server02_UDP.java
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Server02_UDP
```

```
Server is running...
Receive from Client : 3
Receive from Client : 34
Receive from Client : stop
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>javaC Client02_UDP.java
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Client02_UDP
```

```
Server is connected at InetAddress : localhost/127.0.0.1 and Port at : 3176
```

```
Enter Number for Find Cube : 3
```

```
Cube of Number : 27
```

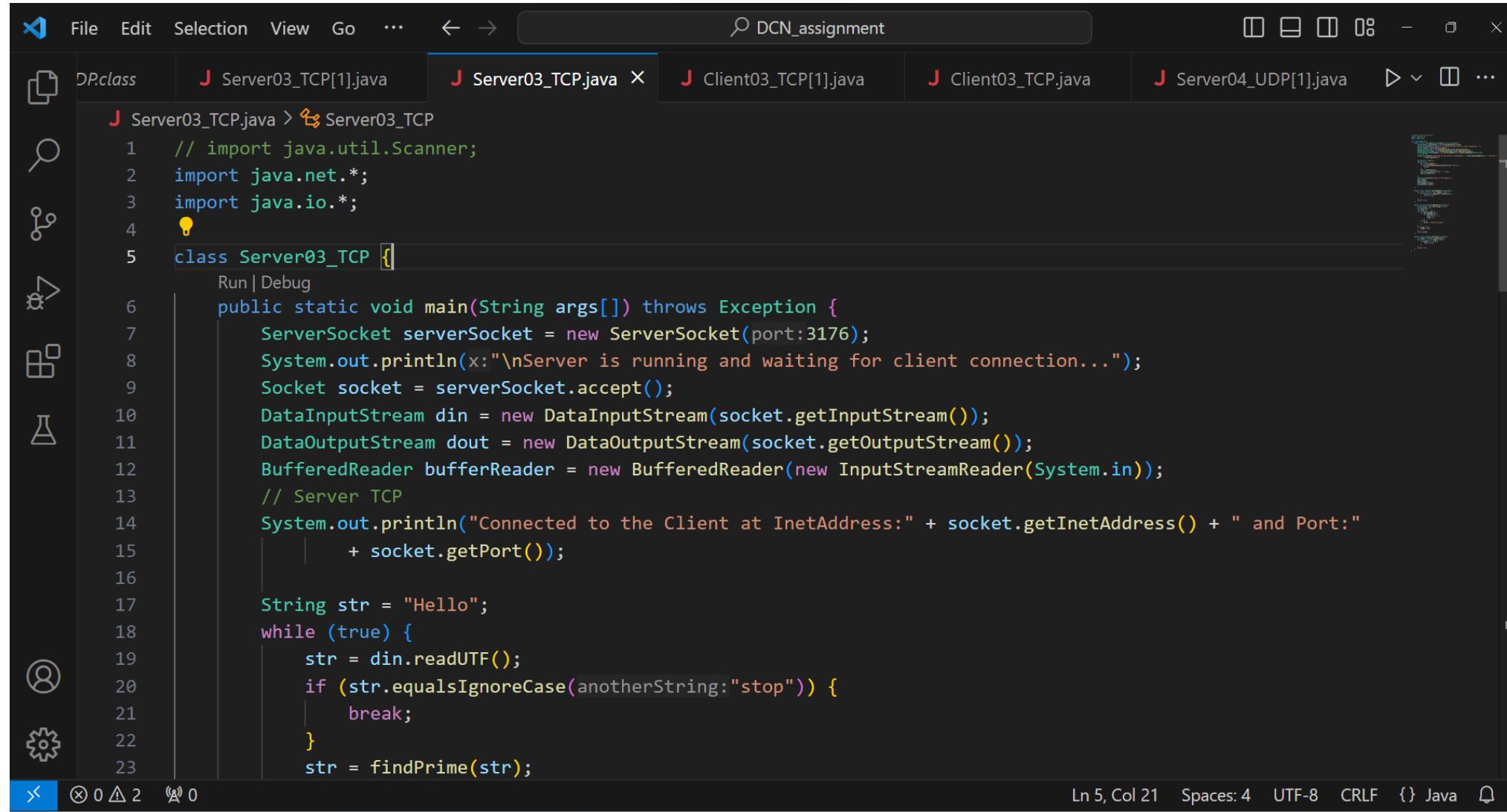
```
Enter Number for Find Cube : 34
```

```
Cube of Number : 39304
```

```
Enter Number for Find Cube : stop
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>
```

q3



The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File, Edit, Selection, View, Go, ...
- Search Bar:** DCN_assignment
- Toolbar:** Includes icons for file operations (New, Open, Save, Find, etc.) and tabs.
- Code Editor:** The active tab is "Server03_TCP.java".

```
1 // import java.util.Scanner;
2 import java.net.*;
3 import java.io.*;
4
5 class Server03_TCP {
6     Run | Debug
7     public static void main(String args[]) throws Exception {
8         ServerSocket serverSocket = new ServerSocket(port:3176);
9         System.out.println(x:"\nServer is running and waiting for client connection...");
10        Socket socket = serverSocket.accept();
11        DataInputStream din = new DataInputStream(socket.getInputStream());
12        DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
13        BufferedReader bufferReader = new BufferedReader(new InputStreamReader(System.in));
14        // Server TCP
15        System.out.println("Connected to the Client at InetAddress:" + socket.getInetAddress() + " and Port:"
16                           + socket.getPort());
17
18        String str = "Hello";
19        while (true) {
20            str = din.readUTF();
21            if (str.equalsIgnoreCase(anotherString:"stop")) {
22                break;
23            }
24            str = findPrime(str);
```
- Status Bar:** Ln 5, Col 21, Spaces: 4, UTF-8, CRLF, {}, Java, Bell icon

The screenshot shows a Java code editor interface with a dark theme. The title bar displays the project name "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The top navigation bar lists several files: DP.class, Server03_TCP[1].java, Server03_TCP.java (which is the active tab), Client03_TCP[1].java, Client03_TCP.java, and Server04_UDP[1].java. The main editor area displays the following Java code:

```
DP.class
Server03_TCP[1].java
Server03_TCP.java X
Client03_TCP[1].java
Client03_TCP.java
Server04_UDP[1].java
D C N_assignment

J Server03_TCP.java > ← Server03_TCP
23     str = findPrime(str);
24     System.out.println("Prime : " + str);
25     dout.writeUTF(str);
26
27 }
28 System.out.println("End of the Program.");
29 din.close();
30 dout.close();
31 bufferedReader.close();
32 serverSocket.close();
33
34 }
35
36 public static boolean verifyNumber(String str) {
37     for (int i = 0; i < str.length(); i++) {
38         if (!Character.isDigit(str.charAt(i))) {
39             return false;
40         }
41     }
42     return true;
43 }
44
45 public static String findPrime(String str) {
46     String set[] = str.split(regex: "\s");
```

The status bar at the bottom shows the current line (Ln 5, Col 21), character count (Spaces: 4), encoding (UTF-8), line endings (CRLF), and file type (Java).

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Explorer:** Shows files: DPclass, Server03_TCP[1].java, Server03_TCP.java (selected), Client03_TCP[1].java, Client03_TCP.java, and Server04_UDP[1].java.
- Code Editor:** Displays the content of `Server03_TCP.java`. The code implements a method to find prime numbers from a set of strings and another method to check if a string is prime.
- Status Bar:** Includes file statistics: Ln 5, Col 21, Spaces: 4, UTF-8, CRLF, and Java.

```
String prime = "";
int flag = 0;
for (String s : set) {
    if (verifyNumber(s)) {
        if (isPrime(s)) {
            prime += s + " ";
            flag = 1;
        }
    } else {
        prime = "Invalid String";
    }
}
if (flag == 0) {
    prime = "0";
}
return prime;
}

public static boolean isPrime(String str) {
    int number = Integer.parseInt(str);
    for (int i = 2; i < number; i++) {
        if (number % i == 0) {
            return false;
        }
    }
}
```

A screenshot of a Java code editor interface. The title bar shows the project name "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The main area displays the content of "Server03_TCP.java".

```
DP.class Server03_TCP[1].java Server03_TCP.java Client03_TCP[1].java Client03_TCP.java Server04_UDP[1].java ...
```

```
Server03_TCP.java > Server03_TCP
53     flag = 1;
54 }
55 } else {
56     prime = "Invalid String";
57 }
58 if (flag == 0) {
59     prime = "0";
60 }
61 return prime;
62 }
63
64
65 public static boolean isPrime(String str) {
66     int number = Integer.parseInt(str);
67     for (int i = 2; i < number; i++) {
68         if (number % i == 0) {
69             return false;
70         }
71     }
72     return true;
73 }
74 }
```

The code implements a prime number checker. It first checks if the input string is valid. If it's not, it returns "Invalid String". If it's zero, it returns "0". Otherwise, it iterates from 2 to the input number-1, checking if any number divides the input evenly. If it finds one, it returns false. If no divisor is found, it returns true.

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Explorer:** Shows files: DPclass, Server03_TCP[1].java, Server03_TCP.java, Client03_TCP[1].java, Client03_TCP.java (selected), and Server04_UDP[1].java.
- Code Editor:** Displays the contents of Client03_TCP.java. The code implements a TCP client to interact with a server. It uses Scanner for input and DataInputStream/DataOutputStream for socket communication.
- Bottom Status Bar:** Shows file statistics: Ln 6, Col 1, Spaces: 4, UTF-8, CRLF, and Java.

```
DPclass
J Server03_TCP[1].java J Server03_TCP.java J Client03_TCP[1].java J Client03_TCP.java X J Server04_UDP[1].java D v ...
```

```
J Client03_TCP.java > ↗ Client03_TCP
1 import java.util.Scanner;
2 import java.net.*;
3 import java.io.*;
4
5 class Client03_TCP {
6     Run | Debug
7     public static void main(String args[]) throws Exception {
8         Socket socket = new Socket(host:"localhost", port:3176);
9         DataInputStream din = new DataInputStream(socket.getInputStream());
10        DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
11        BufferedReader bufferReader = new BufferedReader(new InputStreamReader(System.in));
12        Scanner scanner = new Scanner(System.in);
13        // Client TCP
14        System.out.println("\nConnected to the server at InetAddress:" + socket.getInetAddress() + " and Port:"
15                           + socket.getPort());
16        String str = "";
17        while (true) {
18            System.out.print(s:"\n\nEnter Set of Number ");
19            str = scanner.nextLine();
20            dout.writeUTF(str);
21            if (str.equalsIgnoreCase(anotherString:"stop")) {
22                break;
23            }
}
```

DCN_assignment

DPclass Server03_TCP[1].java Server03_TCP.java Client03_TCP[1].java Client03_TCP.java X Server04_UDP[1].java ...

```
13     System.out.println("Inconnected to the Server at theAddress: " + SOCKET.getInetAddress() + " and PORT: " + socket.getPort());
14
15
16     String str = "";
17     while (true) {
18         System.out.print(":\n\nEnter Set of Number ");
19         str = scanner.nextLine();
20         dout.writeUTF(str);
21         if (str.equalsIgnoreCase("stop")) {
22             break;
23         }
24         dout.flush();
25         str = din.readUTF();
26         System.out.println("Prime Number : " + str);
27     }
28
29     System.out.println("End of the Program.");
30     din.close();
31     dout.close();
32     bufferedReader.close();
33     socket.close();
34     scanner.close();
35 }
36 }
```

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF {} Java

```
C:\Users\MAMTA\Desktop\DCN_assignment>java C Server03_TCP.java
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Server03_TCP
```

```
Server is running and waiting for client connection...
```

```
|
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Client03_TCP.java
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Client03_TCP
```

```
Connected to the server at InetAddress:localhost/127.0.0.1 and Port:3176
```

```
Enter Set of Number 76598
```

```
Prime Number : 0
```

```
Enter Set of Number 34
```

```
Prime Number : 0
```

```
Enter Set of Number 1234
```

```
Prime Number : 0
```

```
Enter Set of Number |
```

q4

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Explorer:** Shows a file tree with Java files: Client03_TCP[1].java, Client03_TCP.java, Server04_UDP[1].java, Server04_UDP.java, and Client04_UDP[1].java.
- Code Editor:** The active tab is Server04_UDP.java. The code implements a UDP server that listens on port 3176. It receives data from clients, converts it to a string, and splits it into integer values. If the received string is "stop", the loop breaks.
- Toolbars and Status Bar:** Includes standard file operations (File, Edit, Selection, View, Go), navigation (Back, Forward), search (Search icon), and status information (Ln 4, Col 19, Spaces: 4, UTF-8, CRLF, Java).

```
import java.net.*;
import java.util.Arrays;
class Server04_UDP {
    public static void main(String args[]) throws Exception {
        DatagramSocket serverSocket = new DatagramSocket(port:3176);
        byte[] receiveData = new byte[1024];
        // Server UDP
        System.out.println(x:"\nServer is running...");
        while (true) {
            DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
            serverSocket.receive(receivePacket);
            String inputData = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
            if (inputData.equalsIgnoreCase(anotherString:"stop")) {
                break;
            }
            String[] strValues = inputData.split(regex:",");
            int[] values = new int[strValues.length];
            for (int i = 0; i < strValues.length; i++) {
```

The screenshot shows a Java code editor interface with the title bar "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, Replace, and Settings. The top navigation bar includes File, Edit, Selection, View, Go, and other standard options. The tabs at the top show several files: Client03_TCP[1].java, Client03_TCP.java, Server04_UDP[1].java 1, Server04_UDP.java X (which is the active tab), and Client04_UDP[1].java C:\. The main code area displays the following Java code:

```
// Sort the values array
Arrays.sort(values);

// Calculate quartiles
double q1 = calculateQuartile(values, percentile:0.25);
double q2 = calculateQuartile(values, percentile:0.5);
double q3 = calculateQuartile(values, percentile:0.75);

// Construct response message
String response = "Q1: " + q1 + ", Q2: " + q2 + ", Q3: " + q3;
byte[] responseData = response.getBytes();

// Get client address and port from received packet
InetAddress clientAddress = receivePacket.getAddress();
int clientPort = receivePacket.getPort();

// Create datagram packet to send response
DatagramPacket sendPacket = new DatagramPacket(responseData, responseData.length, clientAddress,
        clientPort);

// Send the response
serverSocket.send(sendPacket);

}
```

The status bar at the bottom indicates the current line (Ln 4, Col 19), spaces (Spaces: 4), and encoding (UTF-8). It also shows Java and a gear icon.

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File, Edit, Selection, View, Go, ...
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, Select All, Undo, Redo, and Close.
- Code Editor:** The main area displays the `Server04_UDP.java` file with the following content:

```
java J Client03_TCP[1].java J Client03_TCP.java J Server04_UDP[1].java 1 J Server04_UDP.java X J Client04_UDP[1].java C:\ D\ ...  
J Server04_UDP.java > ↗ Server04_UDP  
43     // Create datagram packet to send response  
44     DatagramPacket sendPacket = new DatagramPacket(responseData, responseData.length, clientAddress,  
45             clientPort);  
46  
47     // Send the response  
48     serverSocket.send(sendPacket);  
49 }  
50 serverSocket.close();  
51  
52 }  
53  
54 private static double calculateQuartile(int[] values, double percentile) {  
55     int n = values.length;  
56     double index = percentile * (n - 1) + 1;  
57  
58     if (index % 1 == 0) {  
59         return values[(int) index - 1];  
60     } else {  
61         int k = (int) index;  
62         double fraction = index - k;  
63         return values[k - 1] + fraction * (values[k] - values[k - 1]);  
64     }  
65 }  
66 }
```
- Status Bar:** ShowsLn 4, Col 19 Spaces: 4 UTF-8 CRLF {} Java

The screenshot shows a Java development environment with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File, Edit, Selection, View, Go, ...
- Toolbars:** Standard toolbar icons for file operations.
- Code Editor:** The main window displays the code for `Client04_UDP.java`.

```
1 import java.net.*;
2 import java.util.Scanner;
3
4 class Client04_UDP {
5     public static void main(String args[]) throws Exception {
6         DatagramSocket clientSocket = new DatagramSocket();
7
8         InetAddress serverAddress = InetAddress.getByName(host:"localhost");
9         int serverPort = 3176;
10
11        Scanner scanner = new Scanner(System.in);
12        // Client UDP
13        System.out.println("\nServer is connected at InetAddress : " + serverAddress + " and Port at : " + serverPort);
14
15        while (true) {
16            System.out.print("Enter a set of numeric values separated by commas: ");
17            String inputData = scanner.nextLine();
18
19            byte[] sendData = inputData.getBytes();
20            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);
21            clientSocket.send(sendPacket);
22
23            if (inputData.equalsIgnoreCase(anotherString:"stop")) {
```
- Status Bar:** Shows line 4, column 21, spaces: 4, UTF-8, CRLF, Java, and a bell icon.

The screenshot shows a Java code editor interface with the title bar "DCN_assignment". The left sidebar contains icons for file operations, search, navigation, and other tools. The main workspace displays the code for "Client04_UDP.java".

```
String inputData = scanner.nextLine();

byte[] sendData = inputData.getBytes();
DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);
clientSocket.send(sendPacket);

if (inputData.equalsIgnoreCase(anotherString:"stop")) {
    break;
}

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

String responseData = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
System.out.println("Server Response: " + responseData);

scanner.close();
clientSocket.close();
}
```

The status bar at the bottom indicates the current line (Ln 4), column (Col 21), spaces (Spaces: 4), and encoding (UTF-8). It also shows the file type as Java and includes a notification icon.

```
C:\Users\MAMTA\Desktop\DCN_assignment>java C Server04_UDP.java
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Server04_UDP
```

```
Server is running...
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>
```

```
C:\Windows\System32\cmd.e x + v - o x

Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MAMTA\Desktop\DCN_assignment>java C Client04_UDP.java

C:\Users\MAMTA\Desktop\DCN_assignment>java Client04_UDP

Server is connected at InetAddress : localhost/127.0.0.1 and Port at : 3176
Enter a set of numeric values separated by commas: 10,20,30,
```

q5

File Edit Selection View Go ... 🔍 DCN_assignment

C:\...\1SGZ0FF4 Client04_UDP.java ↗ Server05_UDP[1].java C:\...\1FA86QL4 ↗ Server05_UDP.java ↗ 1 X ↗ Server06_TCP[1].java C:\...

J Server05_UDP.java > ⚙️ Server05_UDP > 📄 main(String[])

```
1 import java.net.*;
2
3 class Server05_UDP {
4     Run | Debug
5     public static void main(String args[]) {
6         try {
7             DatagramSocket serverSocket = new DatagramSocket(port:3176);
8             byte b[] = new byte[1024];
9             // Server UDP
10            System.out.println(x:"\nServer is running...");
11
12            while (true) {
13                DatagramPacket receivePacket = new DatagramPacket(b, b.length);
14                serverSocket.receive(rec | DatagramPacket receivePacket - Server05_UDP.main(String[]))
15                String str = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
16                System.out.println("Receive from Client : " + str);
17
18                String[] set = str.split(regex:",");
19                String value = set[0];
20                String choice = set[1];
21                String response;
22
23                if (verifyNumber(value)) {
24                    if (choice.equals(anObject:"1")) {
```

Ln 10, Col 1 Spaces: 4 UTF-8 CRLF {} Java ⌂

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File List:** C:\...\1SGZ0FF4, Client04_UDP.java, Server05_UDP[1].java C:\...\1FA86QL4, Server05_UDP.java (active), Server06_TCP[1].java C:\...
- Code Editor:** The active tab contains the following Java code:

```
23             if (choice.equals(anObject:"1")) {
24                 response = toFahrenheit(value);
25             } else if (choice.equals(anObject:"2")) {
26                 response = toCelsius(value);
27             } else {
28                 response = "Invalid Input";
29             }
30         } else {
31             response = "Invalid Input";
32         }
33         byte[] responseData = response.getBytes();
34         DatagramPacket sendPacket = new DatagramPacket(responseData, responseData.length,
35             receivePacket.getAddress(), receivePacket.getPort());
36         serverSocket.send(sendPacket);
37     }
38 } catch (Exception e) {
39     e.printStackTrace();
40 }
41 }
42
43 public static boolean verifyNumber(String str) {
44     for (int i = 0; i < str.length(); i++) {
45         if (!Character.isDigit(str.charAt(i))) {
46             return false;
47         }
48     }
49 }
```
- Status Bar:** Ln 45, Col 1, Spaces: 4, UTF-8, CRLF, {}, Java, Bell icon

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File List:** C:\...\1SGZOFF4, Client04_UDP.java, Server05_UDP[1].java C:\...\1FA86QL4, Server05_UDP.java (active), Server05_UDP.java C:\, Server06_TCP[1].java C:\
- Code Area:** The active tab contains the following Java code:

```
42     public static boolean verifyNumber(String str) {
43         for (int i = 0; i < str.length(); i++) {
44             if (!Character.isDigit(str.charAt(i))) {
45                 return false;
46             }
47         }
48         return true;
49     }
50
51
52     public static String toFahrenheit(String str) {
53         float number = Float.parseFloat(str);
54         float result = (float) ((9.0 / 5.0) * number) + 32;
55         return String.valueOf(result);
56     }
57
58     public static String toCelsius(String str) {
59         float number = Float.parseFloat(str);
60         float result = (float) ((number - 32) * (5.0 / 9.0));
61         return String.valueOf(result);
62     }
63 }
64 }
```

The code implements utility methods for validating numbers and converting temperatures between Fahrenheit and Celsius.

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File Edit Selection View Go ...
- Toolbars:** Standard toolbar icons for File, Edit, Selection, View, Go, and others.
- Code Area:** The main area displays the code for `Client05_UDP.java`.

```
1 import java.io.*;
2 import java.util.Scanner;
3 import java.net.*;
4
5 class Client05_UDP {
6     Run | Debug
7     public static void main(String args[]) {
8         try {
9             DatagramSocket clientSocket = new DatagramSocket();
10            InetAddress address = InetAddress.getByName(host:"localhost");
11            int port = 3176;
12            Scanner scanner = new Scanner(System.in);
13            // Client UDP
14            System.out.println("\nServer is connected at InetAddress : " + address + " and Port at : " + port);
15            while (true) {
16                System.out.println(x:"\n\n1 for Celsius to Fahrenheit\n2 for Fahrenheit to Celsius\n3 for Exit");
17                System.out.print(s:"Enter Choice : ");
18                String ch = scanner.next();
19
20                if (ch.equals(anObject:"3")) {
21                    break;
22                }
23
24                System.out.print(s:"Enter Temperature : ");
```
- Sidebar:** A vertical sidebar on the left contains icons for file operations (New, Open, Save, Find, Replace, Cut, Copy, Paste, Delete), a search icon, a refresh icon, a run/debug icon, a file tree icon, a help icon, and a settings gear icon.
- Status Bar:** Shows line 2, column 26, spaces 4, UTF-8, CRLF, Java, and a copy/paste icon.

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File, Edit, Selection, View, Go, ...
- Search Bar:** DCN_assignment
- Toolbars:** Standard toolbar icons (New, Open, Save, Print, Find, Copy, Paste, etc.)
- Code Area:** The main area displays the code for `Client05_UDP.java`. The code handles user input for temperature conversion between Celsius and Fahrenheit.

```
20         break;
21     }
22
23     System.out.print("Enter Temperature : ");
24     String str = scanner.next();
25     str = str + "," + ch;
26     byte b[] = str.getBytes();
27
28     DatagramPacket sendPacket = new DatagramPacket(b, b.length, address, port);
29     clientSocket.send(sendPacket);
30
31     byte b1[] = new byte[1024];
32     DatagramPacket receivePacket = new DatagramPacket(b1, b1.length);
33     clientSocket.receive(receivePacket);
34     str = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
35
36     if (ch.equals(anObject:"1")) {
37         System.out.println("Celsius to Fehrenheit is : " + str);
38     } else if (ch.equals(anObject:"2")) {
39         System.out.println("Fehrenheit to Celsius is : " + str);
40     }
41
42 }
43 scanner.close();
```

- Status Bar:** Ln 2, Col 26, Spaces: 4, UTF-8, CRLF, {} Java, Bell icon

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Menu:** File, Edit, Selection, View, Go, ...
- Toolbars:** Standard toolbar icons for file operations.
- Code Area:** The main area displays the code for `Client05_UDP.java`. The code handles UDP communication between a client and a server, performing unit conversion (Celsius to Fahrenheit or vice versa) based on user input.

```
27     DatagramPacket sendPacket = new DatagramPacket(b, b.length, address, port);
28     clientSocket.send(sendPacket);
29
30     byte b1[] = new byte[1024];
31     DatagramPacket receivePacket = new DatagramPacket(b1, b1.length);
32     clientSocket.receive(receivePacket);
33     str = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
34
35     if (ch.equals(anObject:"1")) {
36         System.out.println("Celsius to Fehrenheit is : " + str);
37     } else if (ch.equals(anObject:"2")) {
38         System.out.println("Fehrenheit to Celsius is : " + str);
39     }
40
41     }
42
43     scanner.close();
44     clientSocket.close();
45 } catch (Exception e) {
46     e.printStackTrace();
47 }
48
49 }
```

- Status Bar:** Shows line 2, column 26, spaces: 4, encoding: UTF-8, CRLF, Java, and a bell icon.

```
C:\Users\MAMTA\Desktop\DCN_assignment>java C Server05_UDP.java
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Server05_UDP
```

```
Server is running...
```

```
Receive from Client : 678,1
```

```
Receive from Client : 5436,2
```

```
C:\Windows\System32\cmd.exe + ▾ - X

Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MAMTA\Desktop\DCN_assignment>javaC Client05_UDP.java

C:\Users\MAMTA\Desktop\DCN_assignment>java Client05_UDP

Server is connected at InetSocketAddress : localhost/127.0.0.1 and Port at : 3176

1 for Celsius to Fahrenheit
2 for Fahrenheit to Celsius
3 for Exit
Enter Choice : 1
Enter Temperature : 678
Celsius to Fehrenheit is : 1252.4

1 for Celsius to Fahrenheit
2 for Fahrenheit to Celsius
3 for Exit
Enter Choice : 2
Enter Temperature : 5436
Fehrenheit to Celsius is : 3002.2222

1 for Celsius to Fahrenheit
2 for Fahrenheit to Celsius
3 for Exit
Enter Choice : 3

C:\Users\MAMTA\Desktop\DCN_assignment>
```

q6

The screenshot shows a Java code editor interface with a dark theme. The title bar reads "DCN_assignment". The left sidebar features a vertical toolbar with icons for file operations, search, navigation, run/debug, and other developer tools. The main workspace displays a Java file named "Server06_TCP.java". The code implements a TCP server that listens on port 3176, accepts connections from clients, and processes requests using DataInputStream and DataOutputStream. It includes logic to handle a specific request for a circle area calculation.

```
1 import java.io.*;
2 import java.net.*;
3 
4 public class Server06_TCP {
5     public static void main(String args[]) {
6         try {
7             ServerSocket serverSocket = new ServerSocket(port:3176);
8             System.out.println("nServer is running and waiting for client connection...");
9             Socket socket = serverSocket.accept();
10            System.out.println("nConnected to the Client at InetAddress:" + socket.getInetAddress() + " and Port
11                           + socket.getPort());
12 
13            DataInputStream input = new DataInputStream(socket.getInputStream());
14            DataOutputStream output = new DataOutputStream(socket.getOutputStream());
15 
16            String area;
17            while (true) {
18                String ch_value = input.readUTF();
19                String[] set = ch_value.split(regex:",");
20                String choice = set[0];
21 
22                if (choice.equals(anObject:"circle")) {
23                    area = areaOfCircle(set[1]);
```

Ln 3, Col 1 Spaces: 4 UTF-8 CRLF {} Java

The screenshot shows a Java code editor interface with a dark theme. The title bar reads "DCN_assignment". The left sidebar features a vertical column of icons: a file folder, a magnifying glass, a gear, a person, a square, a triangle, a rectangle, and a gear. The main workspace displays the following Java code:

```
23     area = areaOfCircle(set[1]);
24 } else if (choice.equals(anObject:"square")) {
25     area = areaOfSquare(set[1]);
26 } else if (choice.equals(anObject:"rectangle")) {
27     area = areaOfRectangle(set[1], set[2]);
28 } else if (choice.equals(anObject:"triangle")) {
29     area = areaOfTriangle(set[1], set[2], set[3]);
30 } else {
31     break;
32 }
33
34     output.writeUTF(area);
35 }
36     socket.close();
37 // serverSocket.close();
38 } catch (Exception e) {
39     e.printStackTrace();
40 }
41 }
42
43 public static String areaOfCircle(String r) {
44     double radius = Double.parseDouble(r);
45     double area = Math.PI * radius * radius;
46     return Double.toString(area);
```

The status bar at the bottom indicates "Ln 3, Col 1" and "Spaces: 4".

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Tabs:** C:\...\1FA86QL4, J Server05_UDP.java \ 1, J Server06_TCP[1].java C:\...\1SGZ0FF4, J Server06_TCP.java \ 1 X, Untitled-3, J Ser D v, ...
- Left Sidebar:** Includes icons for Copy, Find, Open, Save, and Settings.
- Code Area:** Displays the following Java code:

```
47     }
48
49     public static String areaOfSquare(String l) {
50         double length = Double.parseDouble(l);
51         double area = length * length;
52         return Double.toString(area);
53     }
54
55     public static String areaOfRactangle(String l, String b) {
56         double length = Double.parseDouble(l);
57         double width = Double.parseDouble(b);
58         double area = length * width;
59         return Double.toString(area);
60     }
61
62     public static String areaOfTriangle(String a, String b, String c) {
63         double first = Double.parseDouble(a);
64         double second = Double.parseDouble(b);
65         double third = Double.parseDouble(c);
66         double s = (first + second + third) / 2;
67         double area = Math.sqrt((s) * (s - first) * (s - second) * (s - third));
68         return Double.toString(area);
69     }
```

Status Bar: Ln 3, Col 1 Spaces: 4 UTF-8 CRLF {} Java

File Edit Selection View Go ... ← → ⌂ DCN_assignment

J Client06_TCP.java X J Server07_TCP[1].java J Server07_TCP.java J Client07_TCP[1].java J Client07_TCP.java J Server08 ▷ ...

J Client06_TCP.java > ↗ Client06_TCP

17 System.out.println(
18 | "x:\n\n1 : Area of Circle\n2 : Area of Square\n3 : Area of Rectangle\n4 : Area of Triangl
19 System.out.print(s:"Enter Choice : ");
20 String ch = scanner.next();
21 String value, ch_value;
22 if (ch.equals(anObject:"1")) {
23 System.out.println(x:"\n***Area of Triangle***");
24 System.out.print(s:"Enter Radius : ");
25 value = scanner.next();
26 ch_value = "circle," + value;
27 } else if (ch.equals(anObject:"2")) {
28 System.out.println(x:"\n***Area of Square***");
29 System.out.print(s:"Enter Length : ");
30 value = scanner.next();
31 ch_value = "square," + value;
32 } else if (ch.equals(anObject:"3")) {
33 System.out.println(x:"\n***Area of Rectangle***");
34 System.out.print(s:"Enter Length : ");
35 value = scanner.next();
36 System.out.print(s:"\nEnter Width : ");
37 value = value + "," + scanner.next();
38 ch_value = "rectangle," + value;
39 } else if (ch.equals(anObject:"4")) {
System.out.println(x:"\n***Area of Triangle***").

Ln 5, Col 26 Spaces: 4 UTF-8 CRLF {} Java

File Edit Selection View Go ... ← → 🔍 DCN_assignment

J Client06_TCP.java X J Server07_TCP[1].java J Server07_TCP.java J Client07_TCP[1].java J Client07_TCP.java J Server08 D ...

J Client06_TCP.java > Client06_TCP

17 System.out.println(
18 | | x:"\n\n1 : Area of Circle\n2 : Area of Square\n3 : Area of Rectangle\n4 : Area of Triangl
19 System.out.print(s:"Enter Choice : ");
20 String ch = scanner.next();
21 String value, ch_value;
22 if (ch.equals(anObject:"1")) {
23 | | System.out.println(x:"\n***Area of Triangle***");
24 | | System.out.print(s:"Enter Radius : ");
25 | | value = scanner.next();
26 | | ch_value = "circle," + value;
27 } else if (ch.equals(anObject:"2")) {
28 | | System.out.println(x:"\n***Area of Square***");
29 | | System.out.print(s:"Enter Length : ");
30 | | value = scanner.next();
31 | | ch_value = "square," + value;
32 } else if (ch.equals(anObject:"3")) {
33 | | System.out.println(x:"\n***Area of Rectangle***");
34 | | System.out.print(s:"Enter Length : ");
35 | | value = scanner.next();
36 | | System.out.print(s:"\nEnter Width : ");
37 | | value = value + "," + scanner.next();
38 | | ch_value = "rectangle," + value;
39 } else if (ch.equals(anObject:"4")) {
40 | | System.out.println(x:"\n***Area of Triangle***").

Ln 5, Col 26 Spaces: 4 UTF-8 CRLF {} Java

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Explorer:** Shows files: Client06_TCP.java (selected), Server07_TCP[1].java, Server07_TCP.java, Client07_TCP[1].java, Client07_TCP.java, Server08.java.
- Sidebar Icons:** Document, Search, Find, Go To, Open, Save, Run, Stop, Help.
- Code Area:** Displays the following Java code:

```
40     System.out.println(x:"\n***Area of Triangle***");
41     System.out.print(s:"Enter Length of First Side: ");
42     value = scanner.nextInt();
43     System.out.print(s:"\nEnter Length of Second Side: ");
44     value = value + "," + scanner.nextInt();
45     System.out.print(s:"\nEnter Length of Third Side: ");
46     value = value + "," + scanner.nextInt();
47     ch_value = "triangle," + value;
48 } else {
49     break;
50 }
51
52 output.writeUTF(ch_value);
53 output.flush();
54
55 String area = input.readUTF();
56 System.out.println("\n*** Area : " + area);
57 }
58 scanner.close();
59 s.close();
60 }
61
62 catch (Exception e) {
63     e.printStackTrace();
```

Status Bar: Ln 5, Col 26 Spaces: 4 UTF-8 CRLF {} Java

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Explorer:** Shows files: Client06_TCP.java, Server07_TCP[1].java, Server07_TCP.java, Client07_TCP[1].java, Client07_TCP.java, and Server08.java.
- Code Editor:** Displays the content of Client06_TCP.java.

```
43     System.out.print(s:"\nEnter Length of Second Side: ");
44     value = value + "," + scanner.next();
45     System.out.print(s:"\nEnter Length of Third Side: ");
46     value = value + "," + scanner.next();
47     ch_value = "triangle," + value;
48 } else {
49     break;
50 }
51
52     output.writeUTF(ch_value);
53     output.flush();
54
55     String area = input.readUTF();
56     System.out.println("\n*** Area : " + area);
57 }
58 scanner.close();
59 s.close();
60 }
61
62 catch (Exception e) {
63     e.printStackTrace();
64 }
65 }
66 }
```
- Status Bar:** Shows line 5, column 26; spaces: 4; encoding: UTF-8; CRLF; Java; and a search icon.

q7

The screenshot shows a Java code editor interface with the title bar "DCN_assignment". The left sidebar contains icons for file operations, search, navigation, and other development tools. The main area displays a Java file named "Server07_TCP.java". The code implements a TCP server that listens on port 3176, accepts client connections, and processes incoming data using a camelCase variable name. The code editor shows syntax highlighting for Java keywords and comments.

```
1 import java.io.*;
2 import java.net.*;
3
4 public class Server07_TCP {
5     public static void main(String args[]) {
6         try {
7             ServerSocket serverSocket = new ServerSocket(port:3176);
8             System.out.println("Server is running and waiting for client connection...");
9             Socket socket = serverSocket.accept();
10            // Server TCP
11            System.out.println("Connected to the Client at InetAddress:" + socket.getInetAddress() + " and Port:"
12                           + socket.getPort());
13
14            DataInputStream input = new DataInputStream(socket.getInputStream());
15            DataOutputStream output = new DataOutputStream(socket.getOutputStream());
16
17            while (true) {
18                String value = input.readUTF();
19                String[] set = value.split(regex: "\s");
20                String camelCase = "";
21
22                for (String s : set) {
23                    camelCase += Character.toUpperCase(s.charAt(index:0));
```

Ln 2, Col 19 Spaces: 4 UTF-8 CRLF {} Java

A screenshot of a Java code editor interface. The title bar shows the project name "DCN_assignment". The tab bar contains six tabs, with the third one, "Server07_TCP.java", currently selected. The code editor displays the following Java code:

```
String camelCase = "";
for (String s : set) {
    camelCase += Character.toUpperCase(s.charAt(0));
    for (int i = 1; i < s.length(); i++) {
        camelCase += Character.toLowerCase(s.charAt(i));
    }
    camelCase += " ";
}
System.out.println("Output for Send : " + camelCase);
output.writeUTF(camelCase);
if (value.equalsIgnoreCase(anotherString:"stop")) {
    break;
}
socket.close();
serverSocket.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

The code implements a camel case conversion logic. It iterates over a set of strings, concatenating each string's first character in uppercase followed by all subsequent characters in lowercase, separated by a space. It then prints the resulting camel case string to standard output and writes it to a socket. If the value "stop" is received, it breaks out of the loop. Finally, it closes the client and server sockets.

The screenshot shows a Java code editor interface with a dark theme. The title bar displays the project name "DCN_assignment". The left sidebar features a vertical toolbar with icons for file operations, search, navigation, and other development tools. The main workspace contains the code for the class `Client07_TCP`. The code uses standard Java libraries for socket communication and user input. A yellow warning icon is present in the margin next to line 4.

```
1 import java.util.Scanner;
2 import java.net.*;
3 import java.io.*;
4 
5 public class Client07_TCP {
6     Run | Debug
7     public static void main(String[] args) {
8         try {
9             Socket s = new Socket(host:"localhost", port:3176);
10            DataInputStream input = new DataInputStream(s.getInputStream());
11            DataOutputStream output = new DataOutputStream(s.getOutputStream());
12            Scanner scanner = new Scanner(System.in);
13            // Client TCP
14            System.out.println(
15                "Connected to the server at InetAddress:" + s.getInetAddress() + " and Port:" + s.getPort());
16            while (true) {
17                System.out.println(x:"\n***Camel Case***");
18                System.out.print(s:"Enter String : ");
19                String str = scanner.nextLine();
20
21                output.writeUTF(str);
22                output.flush();
23        }
```

Bottom status bar: Ln 5, Col 26 (12 selected) Spaces: 4 UTF-8 CRLF {} Java

The screenshot shows a Java code editor interface with a dark theme. The title bar reads "DCN_assignment". The left sidebar contains icons for file operations like copy, paste, search, and refresh. The top navigation bar has tabs for "File", "Edit", "Selection", "View", "Go", and "Client07_TCP.java". Below the tabs, there are several other tabs: "CP.java", "Server07_TCP[1].java", "Server07_TCP.java", "Client07_TCP[1].java", "Client07_TCP.java X", "Server08_UDP[1].java", and "Client07_TCP...". The main code editor area displays the following Java code:

```
17     System.out.println(x:"\n***Camel Case***");
18     System.out.print(s:"Enter String : ");
19     String str = scanner.nextLine();
20
21     output.writeUTF(str);
22     output.flush();
23
24     str = input.readUTF();
25     System.out.println("String After Modification : " + str);
26
27     if (str.equalsIgnoreCase(anotherString:"stop")) {
28         break;
29     }
30
31     scanner.close();
32     s.close();
33
34
35     catch (Exception e) {
36         e.printStackTrace();
37     }
38
39 }
```

The status bar at the bottom shows "Ln 5, Col 26 (12 selected)" and "Spaces: 4".

```
C:\Windows\System32\cmd.exe + v - 0 X

Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MAMTA\Desktop\DCN_assignment>javac Client07.java
error: file not found: Client07.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\MAMTA\Desktop\DCN_assignment>javac Client07_TCP.java

C:\Users\MAMTA\Desktop\DCN_assignment>java Client07_TCP
Connected to the server at InetAddress:localhost/127.0.0.1 and Port:3176

***Camel Case***
Enter String : hello DCN
String After Modification : Hello Dcn
```

```
C:\Windows\System32\cmd.exe + v Microsoft Windows [Version 10.0.22621.3007]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\MAMTA\Desktop\DCN_assignment>java C Server07_TCP.java  
  
C:\Users\MAMTA\Desktop\DCN_assignment>java Server07_TCP  
  
Server is running and waiting for client connection...  
Connected to the Client at InetSocketAddress:127.0.0.1 and Port:51913  
Output for Send : Hello Dcn  
Output for Send : Stop  
  
C:\Users\MAMTA\Desktop\DCN_assignment>
```

q8

The screenshot shows a Java code editor interface with a dark theme. The title bar reads "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The top navigation bar has tabs for ".java", "Client07_TCP[1].java", "Client07_TCP.java", "Server08_UDP[1].java", "Server08_UDP.java", "Client08_UDP[1].java", and "Run | Debug". The main editor area displays the following Java code:

```
import java.net.*;
class Server08_UDP {
    public static void main(String args[]) {
        try {
            DatagramSocket serverSocket = new DatagramSocket(port:3176);
            byte b[] = new byte[1024];
            String str = "";
            // Server UDP
            System.out.println("Server is running...");
            while (true) {
                DatagramPacket receivePacket = new DatagramPacket(b, b.length);
                serverSocket.receive(receivePacket);
                str = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
                String[] set = str.split(regex:"\\s");
                String camelCase = "";
                for (String s : set) {
                    camelCase += Character.toLowerCase(s.charAt(index:0));
                    for (int i = 1; i < s.length(); i++) {
                        camelCase += Character.toUpperCase(s.charAt(i));
                    }
                }
            }
        }
    }
}
```

The status bar at the bottom shows "Ln 4, Col 1" and "Spaces: 4" and "UTF-8". There are also icons for CRLF, Java, and a bell.

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File List:** :P.java, Client07_TCP[1].java, Client07_TCP.java, Server08_UDP[1].java, Server08_UDP.java (highlighted), Client08_UDP[1].java
- Code Area:** The current file is `Server08_UDP.java`. The code implements a UDP server that converts user input to camelCase and sends it back. It includes error handling for a "stop" command.

```
for (String s : set) {
    camelCase += Character.toLowerCase(s.charAt(0));
    for (int i = 1; i < s.length(); i++) {
        camelCase += Character.toUpperCase(s.charAt(i));
    }
    camelCase += " ";
}
System.out.println("Output for Send : " + camelCase);
b = camelCase.getBytes();
DatagramPacket sendPacket = new DatagramPacket(b, b.length, receivePacket.getAddress(),
                                                receivePacket.getPort());
serverSocket.send(sendPacket);
if (camelCase.equalsIgnoreCase(anotherString:"stop")) {
    break;
}
serverSocket.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

- Sidebar:** On the left, there is a vertical sidebar with various icons: a file icon, a search icon, a connection icon, a refresh icon, a grid icon, a triangle icon, a user icon, and a gear icon.
- Status Bar:** At the bottom, the status bar displays: Ln 4, Col 1, Spaces: 4, UTF-8, CRLF, {}, Java, and a help icon.

The screenshot shows a Java code editor interface with a dark theme. The title bar reads "DCN_assignment". The left sidebar contains icons for file operations like Open, Save, Find, and Run. The top navigation bar includes "File", "Edit", "Selection", "View", "Go", and "..." followed by a search bar and window control buttons.

The code editor displays the contents of the file "Client08_UDP.java". The code is as follows:

```
java J Server08_UDP[1].java J Server08_UDP.java J Client08_UDP[1].java J Client08_UDP.java X J Client04_UDP[1].java C:\ ▶ ⌂ ...
```

```
J Client08_UDP.java > ↗ Client08_UDP
1 import java.net.*;
2 import java.util.Scanner;
3
4 public class Client08_UDP {
    Run | Debug
5     public static void main(String args[]) {
6         try {
7             DatagramSocket clientSocket = new DatagramSocket();
8             byte b[] = new byte[1024];
9
10            Scanner scanner = new Scanner(System.in);
11            InetAddress address = InetAddress.getByName(host:"localhost");
12            int port = 3176;
13            String str;
14            // Client UDP
15            System.out.println("Server is connected at InetAddress : " + address + " and Port at : " + port);
16
17            while (true) {
18                System.out.println(x:"\n***Toggle Case***");
19                System.out.print(s:"Enter String : ");
20                str = scanner.nextLine();
21                b = str.getBytes();
22                DatagramPacket sendPacket = new DatagramPacket(b, b.length, address, port);
23                clientSocket.send(sendPacket);
```

The status bar at the bottom shows "Ln 5, Col 1" and other settings like "Spaces: 4", "UTF-8", "CRLF", and "Java".

The screenshot shows a Java code editor interface with the following details:

- Title Bar:** DCN_assignment
- File Explorer:** Shows files: Server08_UDP[1].java, Server08_UDP.java, Client08_UDP[1].java, Client08_UDP.java (selected), and Client04_UDP[1].java.
- Code Editor:** Displays the contents of Client08_UDP.java.

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

public class Client08_UDP {
    public static void main(String args[]) {
        try {
            DatagramSocket clientSocket = new DatagramSocket();
            byte b[] = new byte[1024];

            Scanner scanner = new Scanner(System.in);
            InetAddress address = InetAddress.getByName(host:"localhost");
            int port = 3176;
            String str;
            // Client UDP
            System.out.println("Server is connected at InetAddress : " + address + " and Port at : " + port);

            while (true) {
                System.out.println(x:"\n***Toggle Case***");
                System.out.print(s:"Enter String : ");
                str = scanner.nextLine();
                b = str.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(b, b.length, address, port);
                clientSocket.send(sendPacket);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```
- Status Bar:** Ln 5, Col 1 | Spaces: 4 | UTF-8 | CRLF | {} Java | Bell icon

File Edit Selection View Go ... ← → DCN_assignment

java J Server08_UDP[1].java J Server08_UDP.java J Client08_UDP[1].java J Client08_UDP.java X J Client04_UDP[1].java C:\ D ...

J Client08_UDP.java > Client08_UDP

```
17     while (true) {
18         System.out.println("n***Toggle Case***");
19         System.out.print("Enter String : ");
20         str = scanner.nextLine();
21         b = str.getBytes();
22         DatagramPacket sendPacket = new DatagramPacket(b, b.length, address, port);
23         clientSocket.send(sendPacket);
24
25         DatagramPacket receivePacket = new DatagramPacket(b, b.length);
26         clientSocket.receive(receivePacket);
27         str = new String(receivePacket.getData(), offset:0, receivePacket.getLength());
28         System.out.println("String After Modification : " + str);
29         if (str.equalsIgnoreCase(anotherString:"stop")) {
30             break;
31         }
32     }
33     scanner.close();
34     clientSocket.close();
35 } catch (Exception e) {
36     e.printStackTrace();
37 }
38 }
39 }
```

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF {} Java

```
C:\Users\MAMTA\Desktop\DCN_assignment>java C Server08_UDP.java
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Server08_UDP
```

```
Server is running...
```

```
Output for Send : hELLO wORLD
```

```
Output for Send : hI, hERE iS
```

```
Output for Send : hELLO wORLD
```

```
Output for Send : sTOP
```

```
|
```

```
C:\Users\MAMTA\Desktop\DCN_assignment>java Client08_UDP.java

C:\Users\MAMTA\Desktop\DCN_assignment>java Client08_UDP
Server is connected at InetAddress : localhost/127.0.0.1 and Port at : 3176

***Toggle Case***
Enter String : hello world
String After Modification : hELLO wORLD

***Toggle Case***
Enter String : hi, here is mamta
String After Modification : hI, hERE iS

***Toggle Case***
Enter String : hello world
String After Modification : hELLO wORLD

***Toggle Case***
Enter String : stop
String After Modification : sTOP

C:\Users\MAMTA\Desktop\DCN_assignment>
```

thankyou