

**DS - Assignment 3:** MultiThreaded Client-Server Communications using sockets

**MultiThreadedServer.java ->**

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

public class MultiThreadedServer {
    private static final int PORT = 1234;

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

            while (true) {
                // Accept a new client connection
                Socket clientSocket = serverSocket.accept();
                System.out.println("New client connected: " +
clientSocket.getInetAddress().getHostAddress());

                // Create a new thread to handle the client's request
                new ClientHandler(clientSocket).start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

class ClientHandler extends Thread {
    private Socket clientSocket;

    public ClientHandler(Socket socket) {
        this.clientSocket = socket;
    }

    public void run() {
        try (
            // Input stream to receive data from the client
            BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
```

```

        // Output stream to send data to the client
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true)
    ) {
        String message;
        while ((message = in.readLine()) != null) {
            System.out.println("Received message from client: " + message);

            // Reverse the message
            String reversedMessage = new
StringBuilder(message).reverse().toString();

            // Send the reversed message back to the client
            out.println("Reversed message: " + reversedMessage);
            System.out.println("Sent reversed message to client: " +
reversedMessage);
        }
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            // Close the client socket
            clientSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
}
}

```

**Client.java ->**

```

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

public class Client {
    private static final String SERVER_ADDRESS = "localhost";
    private static final int SERVER_PORT = 1234;

    public static void main(String[] args) {
        try (Socket socket = new Socket(SERVER_ADDRESS, SERVER_PORT)) {
            System.out.println("Connected to the server");
        }
    }
}

```

```

        // Create output stream to send messages to the server
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        // Create input stream to receive messages from the server
        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
        // Create input stream to read from console
        BufferedReader consoleInput = new BufferedReader(new
InputStreamReader(System.in));

        String message;
        while (true) {
            System.out.print("Enter a message to send (type 'exit' to
quit): ");
            message = consoleInput.readLine();

            if ("exit".equalsIgnoreCase(message)) {
                break;
            }

            // Send the message to the server
            out.println(message);

            // Receive the reversed message from the server
            String response = in.readLine();
            System.out.println("Server response: " + response);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

## Outputs:->

### Server->

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> javac MultiThreadedServer.java
PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> java MultiThreadedServer
Server is listening on port 1234
New client connected: 127.0.0.1
New client connected: 127.0.0.1
Received message from client: HelloWorld
Sent reversed message to client: dlroWolleH
Received message from client: SanyamAgrawal
Sent reversed message to client: lawargAmaynaS
```

### Client1 ->

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> javac Client.java
PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> java Client
Connected to the server
Enter a message to send (type 'exit' to quit): HelloWorld
Server response: Reversed message: dlroWolleH
Enter a message to send (type 'exit' to quit): exit
PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> 
```

### Client2->

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> javac Client.java
PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> java Client
Connected to the server
Enter a message to send (type 'exit' to quit): SanyamAgrawal
Server response: Reversed message: lawargAmaynaS
Enter a message to send (type 'exit' to quit): exit
PS C:\Users\pc\Desktop\SE21UCSE192_Dis_Sys\Lab3> 
```