Practical – 1

- **AIM:** To implement reliable client-server communication using TCP Socket API. Write a java program where client sends a string as a message and sever counts the characters in the received message from client. Server sends this value back to the client. Server should be able to serve multiple clients simultaneously.

- **Objectives:** To learn client server communication using TCP socket API in java.

- **Theory:**
  - Java Socket programming is used for communication between the applications running on different JRE.
  - Java Socket programming can be connection-oriented or connection-less. Socket and ServerSocket classes are used for connection-oriented socket programming. The client in socket programming must know two information:
  - IP Address of Server, and
  - Port number
  - **Socket class**
    - A socket is simply an endpoint for communications between the machines. The Socket class can be used to create a socket.

```
Socket s = new Socket("localhost",6666); // Code at client side
```

  - **ServerSocket class**
    - The ServerSocket class can be used to create a server socket. This object is used to establish communication with the clients.

```
// Code at server side
ServerSocket ss = new ServerSocket(6666);

//establishes connection and waits for the
client
Socket s=ss.accept();
```

- **Tools / Material Needed:**
  - **Hardware:** Desktop/Laptop with good configuration
  - **Software:** Java, VS Code

- **Procedure:**
  1. Set up `ServerSocket`.
  2. Listen for incoming client connections.
  3. Spawn a new thread for each connection.
  4. Receive messages from client.
  5. If message is not "0", count characters and send count.
  6. Create `Socket` and connect in client.

7. Prompt user for message and send to server.
8. Receive and display server's response.
9. Close all sockets and streams.
10. Ensure server handles multiple clients with threads.

- **Code:**

mserver.java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.io.EOFException;
public class mserver {
    public static void main(String[] args) {
        try(ServerSocket ss = new ServerSocket(5000)){
            System.out.println("Server started");
            while (true) {
                Socket socket = ss.accept();
                System.out.println("New client connected");
                Thread thread = new Thread(() -> {
                    try {
                        DataInputStream data = new DataInputStream(socket.getInputStream());
                        DataOutputStream out = new DataOutputStream(socket.getOutputStream());
                        while (true) {
                            String msg = null;
                            try {
                                msg = data.readUTF();
                            } catch (EOFException e) {
                                System.out.println("Client closed connection");
                                break;
                            }
                            if (msg.equals("0")) {
                                String output = "Connection closed";
                                out.writeUTF(output);
                                System.out.println("Connection closed");
                                break;
                            }
                            String output = "Your Message: " + msg + "\nMessage Length: " + msg.length();
                            out.writeUTF(output);
                        }
                        socket.close();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                    System.out.println("Server closed");
                });
                thread.start();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

mclient.java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
```

```java
import java.net.Socket;
import java.util.Scanner;
public class mclient {
    public static void main(String[] args) {
        try(Socket socket = new Socket("localhost", 5000)) {
            DataInputStream in = new DataInputStream(socket.getInputStream());
            DataOutputStream out = new DataOutputStream(socket.getOutputStream());
            Scanner scanner = new Scanner(System.in);
            while (true) {
                System.out.print("Enter a message (or '0' to close connection): ");
                String message = scanner.nextLine();
                out.writeUTF(message);
                out.flush();
                if (message.equals("0")) {
                    System.out.println("Closing connection...");
                    break;
                }
                String response = in.readUTF();
                System.out.println("Server response: " + response);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

- **Output:**

```
PS D:\Users\Jabarson Christian\Desktop\S6\AJAVA\lab\P1> java mserver
Server started
New client connected
New client connected
Connection closed
Server closed
Connection closed
Server closed
New client connected
Connection closed
Server closed
```

```
PS D:\Users\Jabarson Christian\Desktop\S6\AJAVA\lab\P1> java mclient
Enter a message (or '0' to close connection): Hello!
Server response: Your Message: Hello!
Message Length: 6
Enter a message (or '0' to close connection): How are you?
Server response: Your Message: How are you?
Message Length: 12
Enter a message (or '0' to close connection): 0
Closing connection...
PS D:\Users\Jabarson Christian\Desktop\S6\AJAVA\lab\P1> java mclient
Enter a message (or '0' to close connection): Bye!
Server response: Your Message: Bye!
Message Length: 4
Enter a message (or '0' to close connection): 0
Closing connection...
```

```
PS D:\Users\Jabarson Christian\Desktop\S6\AJAVA\lab\P1> java mclient
Enter a message (or '0' to close connection): HI!
Server response: Your Message: HI!
Message Length: 3
Enter a message (or '0' to close connection): 0
Closing connection...
PS D:\Users\Jabarson Christian\Desktop\S6\AJAVA\lab\P1>
```

Signature of Faculty: **Grade:**