

1. Java version output in the terminal.

```
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$ java -version
openjdk version "21.0.6" 2025-01-21
OpenJDK Runtime Environment (build 21.0.6+7-Ubuntu-124.04.1)
OpenJDK 64-Bit Server VM (build 21.0.6+7-Ubuntu-124.04.1, mixed mode, sharing)
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$
```

2. MyServer.java code in the editor.

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

public class my_server {
    public static void main(String[] args) {
        // Try-with-resources to automatically close resources
        try (
            ServerSocket ss = new ServerSocket(6666); // Create server socket
        ) {
            System.out.println("Server is running and waiting for a client...");

            try (
                Socket s = ss.accept(); // Wait for client connection
                DataInputStream dis = new DataInputStream(s.getInputStream()) // Stream to read from c
            ) {
                System.out.println("Client connected!");

                // Read message from client
                String str = dis.readUTF();
                System.out.println("Message received from client: " + str);
            } catch (IOException e) {
                System.out.println("Error in client communication: " + e);
            }

        } catch (IOException e) {
            System.out.println("Error in server setup or client connection: " + e);
        }
    }
}
```

3. MyClient.java code in the editor.

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

public class my_client {
    public static void main(String[] args) {
        // Try-with-resources to automatically close resources
        try (
            Socket s = new Socket("localhost", 6666); // Connect to server
            DataOutputStream dout = new DataOutputStream(s.getOutputStream()) // Output stream to send data
        ) {
            dout.writeUTF("Hello Server"); // Send message to server
            dout.flush(); // Flush the stream to ensure data is sent
        } catch (IOException e) {
            System.out.println("Error in client communication: " + e);
        }
    }
}
```

4. Command prompt showing successful compilation.

```
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$ javac my_server.java
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$ javac my_client.java
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$
```

5. Server running and waiting for a client.

```
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$ java my_server
Server is running and waiting for a client...
```

6. Client running and sending the message.

```
student@student-VMware-Virtual-Platform:~$ cd "/home/student/Documents/Computer Networks/Exp 8"
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$ java my_client
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$
```

7. Server showing received message.

```
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$ java my_server
Server is running and waiting for a client...
Client connected!
Message received from client: Hello Server
student@student-VMware-Virtual-Platform:~/Documents/Computer Networks/Exp 8$
```