Experiment No. 10

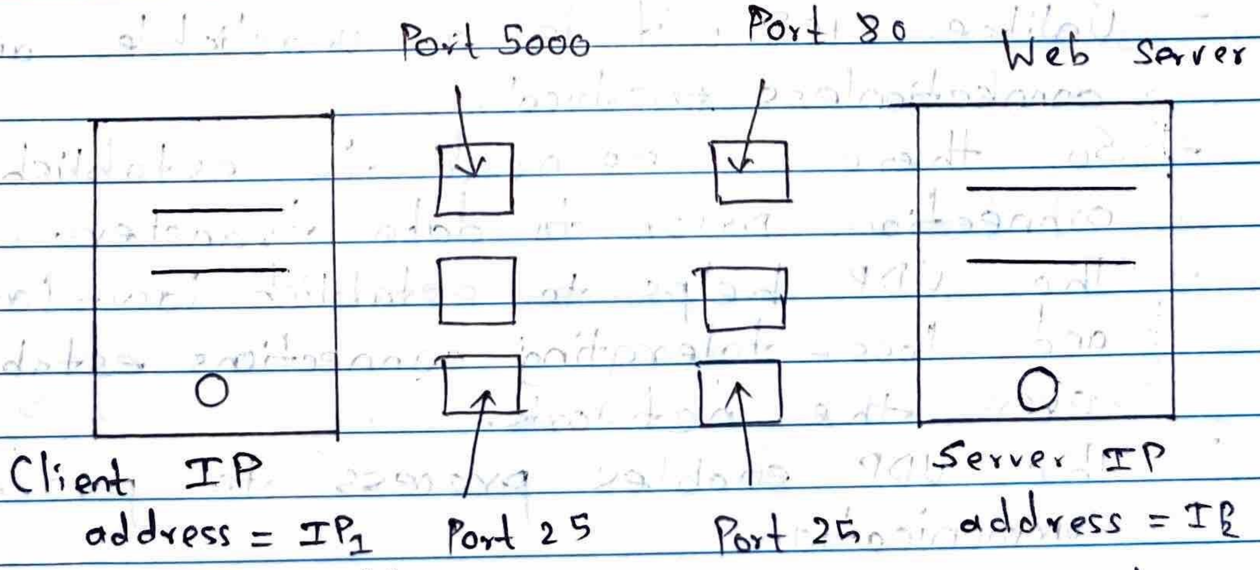
Aim : Socket Programming using TCP or UDP

Theory:

- **Socket**

⇒

- A computer network is a set of devices connected to exchange information and resource such as files, data, images, etc.
- The communication between two or more devices is the communication between the processes present on the different nodes or different computers in a network.
- The communication between different processes on the same nodes or different nodes is done using the concept of a socket.
- A socket is an end-point structure that allows the communication between processes i.e. sending and receiving data over a network.



IP address + Port Number = Socket

- Transmission Control Protocol (TCP)

⇒

- TCP is one of the main protocols of the internet protocol suite.
- It lies between the application and network layers which are used in providing reliable delivery services.
- It is a connection-oriented protocol for communications that helps in the exchange of messages between different devices over a network.
- The IP which establishes the technique for sending data packets between computers, works with TCP.

- User Datagram Protocol (UDP)

⇒

- UDP is a transport layer protocol. UDP is a part of IP suite, referred to as UDP/IP suite.
- Unlike TCP, it is an unreliable and connectionless protocol.
- So there is no need to establish a connection prior to data transfer.
- The UDP helps to establish low-latency and loss-tolerating connections establish over the network.
- The UDP enables process to process communication.

- Creating Server :

- To create the server application, we need to create the instance of Server Socket class.
- Here, we are using 6666 port number for the communication between the client and server.
- The accept () method waits for the client. If the clients connects with the given port number, it returns an instance of socket.

```
ServerSocket   ss   =   ServerSocket (6666);
Sockot   s   =   ss.accept();
```

- Creating Client:

- To create the client application, we need to create the instance of Socket class.
- Here, we need to pass the IP address or hostname of the Server and a port number.
- Here we are using "localhost" because our server is running on same system.

```
Socket  s  =  new  Socket ("localhost", 6666);
```

A 26/09/23.

Code:

(Server.java)

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

public class Server {
    public static void main(String[] args) {
        try {
            ServerSocket ss = new ServerSocket(6666);
            Socket s = ss.accept(); // establishes connection
            DataInputStream dis = new
DataInputStream(s.getInputStream());
            String str = (String) dis.readUTF();
            System.out.println("Message= " + str);
            ss.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

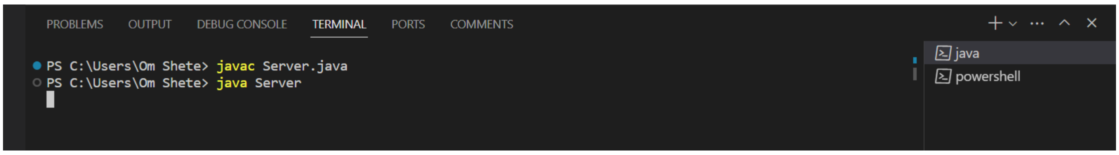
(Client.java)

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

public class Client {
    public static void main(String[] args) {
        try {
            Socket s = new Socket("localhost", 6666);
            DataOutputStream dout = new
DataOutputStream(s.getOutputStream());
            dout.writeUTF("Hello Server");
            dout.flush();
            dout.close();
            s.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

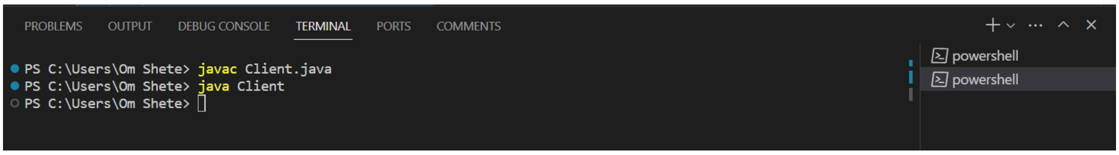
Output:

To execute this program open two command prompts and execute each program at each command prompt as displayed in the below figures. First, run the Server.java file in terminal/cmd,



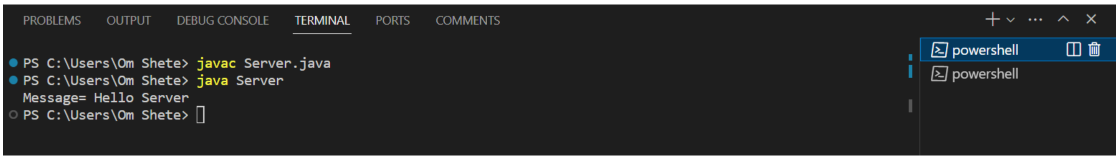
```
● PS C:\Users\Om Shete> javac Server.java
○ PS C:\Users\Om Shete> java Server
```

Running Server.java
Then in the new terminal/cmd run the Client.java file



```
● PS C:\Users\Om Shete> javac Client.java
● PS C:\Users\Om Shete> java Client
○ PS C:\Users\Om Shete> 
```

Running Client.java
As soon as you run the Client program a message is sent to the server and displayed in the Server Terminal/CMD as shown below,



```
● PS C:\Users\Om Shete> javac Server.java
● PS C:\Users\Om Shete> java Server
  Message= Hello Server
○ PS C:\Users\Om Shete> 
```

Message displayed in Server after running Client

CONCLUSION: So, in this experiment, we have successfully understood the concept of Socket Programming and implemented it using Java Programming