East West University
Department of Computer Science and Engineering
A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka

Lab Manual: 09

Lab Topic: Java Socket Programming

Course Code: CSE110 (Object Oriented Programming)

Course Instructor: Mahamudul Hasan, Senior Lecturer, CSE

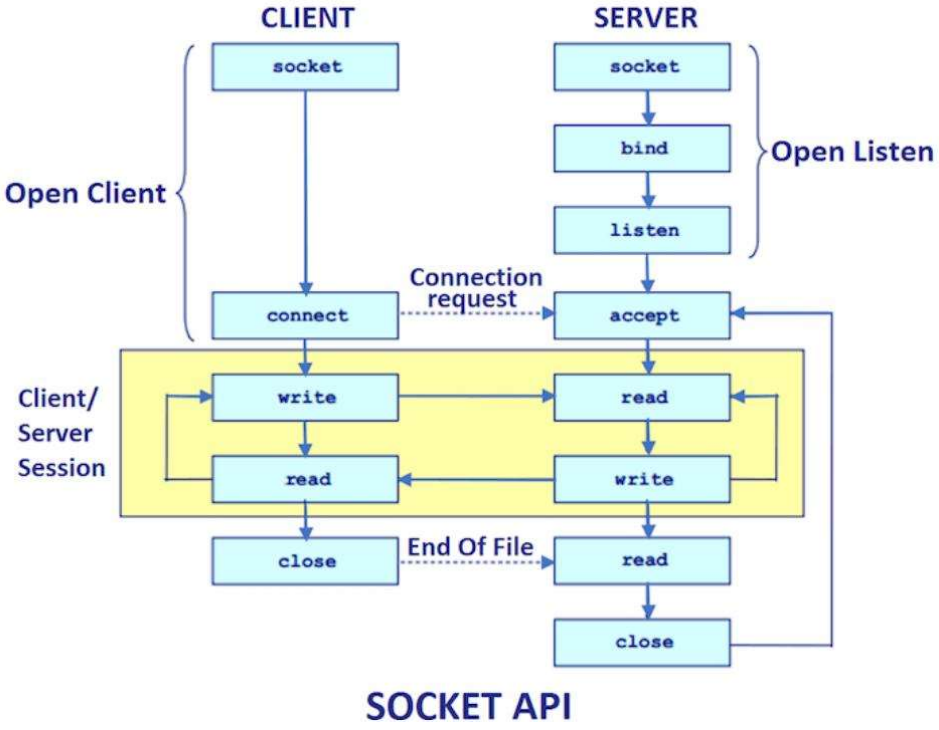
Lab Objective

1. **Understand** the concept of java socket programming
2. **Write** codes to implement connection between client and server.

Lab Activities

A. Java Socket Programming

- 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 and DatagramSocket and DatagramPacket classes are used for connection-less socket programming.
- The client in socket programming must know two information:
    1. IP Address of Server, and
    2. Port number



B. Socket Class

- A socket is simply an endpoint for communications between the machines. The Socket class can be used to create a socket.
- **Important methods**

Method	Description
1) public InputStream getInputStream()	returns the InputStream attached with this socket.
2) public OutputStream getOutputStream()	returns the OutputStream attached with this socket.
3) public synchronized void close()	closes this socket

C. ServerSocket Class

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

Method	Description
1) public Socket accept()	returns the socket and establish a connection between server and client.
2) public synchronized void close()	closes the server socket.

D. Example

Creating Server

- To create the server application, we need to create the instance of ServerSocket class named ***MyServer.java*.**

```
import java.io.*;
import java.net.*;
public class MyServer {
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);}
}
}
```

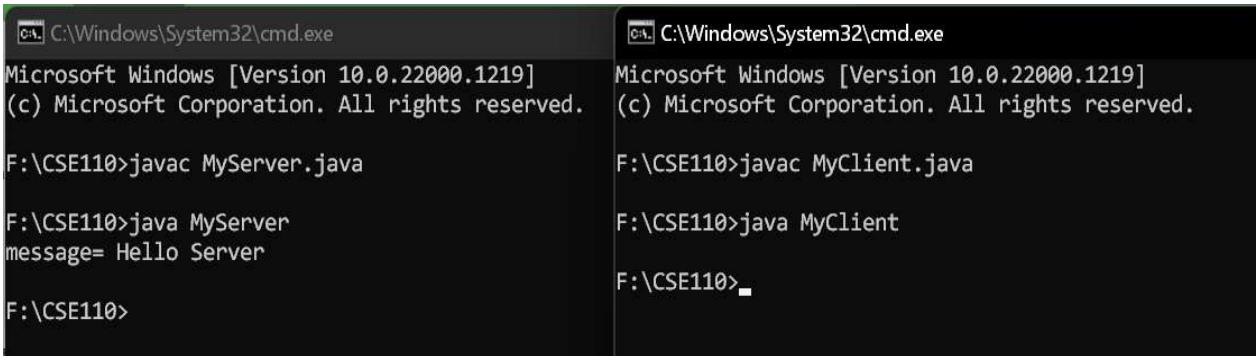
Creating Client

- To create the client application, we need to create the instance of Socket class named ***MyClient.java*.** Here, we need to pass the IP address or hostname of the Server and a port

number.

```
import java.io.*;
import java.net.*;
public class MyClient {
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);}
}
}
```

Sample Output



E. Lab task (Java Socket Programming (Read-Write both side))

1. Create three programs, two of which are clients to a single server. **Client1** will send a character to the server process. The **server** will decrement the letter to the next letter in the alphabet and send the result to the client2. **Client2** prints the letter it receives and then all the processes terminate

2. Next **Client1** will send a random number to the server process. The **server** will check if the number is odd or even and will send the result to **client2**. **Client2** will print the next odd/even number it receives and then all the processes will terminate.