

# Socket programming

Developing a software to work on network. It provides a way of communication between two application or some or different machine.

```
import java.net.*;  
import java.io.*;  
public class Client {
```

```
    private Socket socket = null;  
    private DataInputStream input = null;  
    private DataOutputStream out = null;  
    public Client(String address, int port)  
    {  
        try {
```

```
            socket = new Socket(address, port);  
            System.out.println("connected");  
            input = new DataInputStream(System.in);  
            out = new DataOutputStream(socket.getOutputStream());
```

takes input from terminal.  
sends the data output to the socket

```
        } catch (UnknownHostException u) {  
            System.out.println(u);  
        } catch (IOException i) {  
        }  
    }  
    String line = "";
```

```
    while (!line.equals("Over")) {  
        try {  
            line = input.readLine();  
            out.write(line);  
        } catch (IOException i) {  
        }  
    }  
}
```

Give a simplified example available on next page

1) close client

try {

input.close();

out.close();

socket.close();

}

catch (IOException e) {

}

}

public static void main (String[] args) {

Client client = new Client("127.0.0.1", 5000);

}

}

import java.net.\*;

import java.io.\*;

public class Server {

private Socket socket = null;

private ServerSocket server = null;

private DataInputStream in = null;

public Server(int port) {

try {

server = new ServerSocket(port);

System.out.println("waiting for client");

System.out.println("Server started");

socket = server.accept();

in = new DataInputStream(

new BufferedInputStream(

socket.getInputStream());

```
String line = "";  
while (!line.equals("over"))  
{  
    try {  
        line = in.readLine();  
        System.out.println(line);  
    }  
    catch (IOException i)  
    {  
        System.out.println(i);  
    }  
}
```

```
System.out.println("closing connection");  
socket.close();  
in.close();
```

```
catch (Exception i) {  
}
```

```
}  
public static void main (String[] args)
```

```
{  
    Server server = new Server(1234);
```

```
}  
}
```

```

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

public class SocketClient {
    public static void main(String[] args) {
        String ip = "localhost";
        int port = 5000;
        Socket sock = new Socket(ip, port);
        String str = "HELLO World";
        OutputStreamWriter os = new OutputStreamWriter(sock);
        OutputStreamWriter os = new OutputStreamWriter(sock.getOutputStream());
        PrintWriter out = new PrintWriter(os);
        out.println(str);
        os.flush();
    }
}

```

Server example next page

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

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

```
public class Server {
```

```
    public static void main(String[] args) {
```

```
        System.out.println("Server is started");
```

```
        ServerSocket ssock = new ServerSocket(5000);
```

```
        System.out.println("Server is waiting for client request");
```

```
        Socket sc = ssock.accept();
```

```
        System.out.println("Server is connected");
```

```
        BufferedReader br = new BufferedReader(  
            new InputStreamReader(sc.getInputStream());
```

```
        String str = br.readLine();
```

```
        System.out.println(str);
```

```
    }
```

```
}
```