

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY



DEPARTMENT OF ICT

Lab Report No : 03

Course Code : ICT-3208

Course Title : Network Planning and Designing Lab

Lab Report Name : Socket Programming Lab

<i>Submitted by</i>	<i>Submitted to</i>
Name: Ashik Mahmud ID : IT-17009 Session : 2016-2017 3rd Year 2 nd Semester	Nazrul Islam Assistant Professor, Department of ICT, MBSTU Santosh, Tangail-1902

Date of Submission : 23rd September, 2020

Socket Programming:

Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming.

Socket connections normally run between two different computers on a Local Area Network(LAN) or across the internet, but they can also be used for interprocess communication on a single computer.

The client in socket programming must know two information:

- 1.IP Address of Server, and
- 2.Port number.

ClientSide Source code :

```
package socket;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.Socket;
public class SocketClient {
    public static void main(String[] args) throws IOException {
        String ip = "localhost";
        int port = 8090;
        Socket socket = new Socket(ip, port);
        String msg = "Md. Shoriful Islam Sakib";
        System.out.println("Hello server, "+msg);
        OutputStreamWriter os = new
OutputStreamWriter(socket.getOutputStream());
        PrintWriter out = new PrintWriter(os);
        os.write(msg);
        os.flush();
    }
}
```

Output:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home.
Hello server, Md. Shoriful Islam Sakib

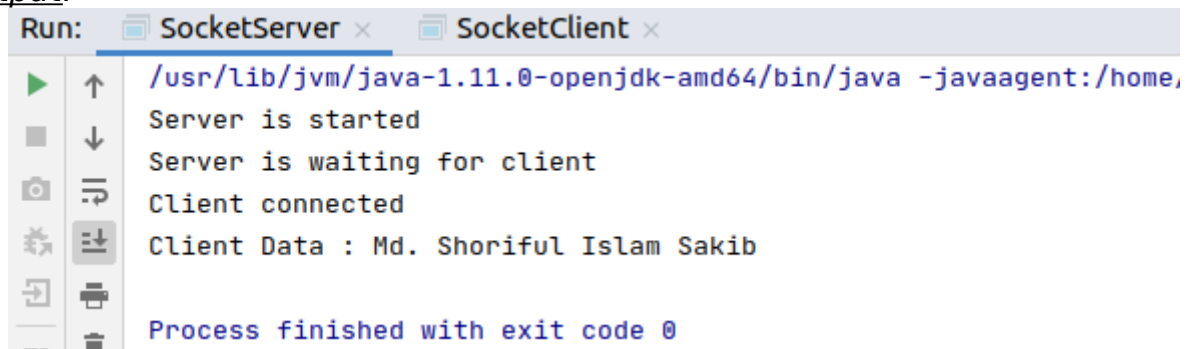
Process finished with exit code 0
```

ServerSide Source code:

```
package socket;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;
public class SocketServer {
```

```
public static void main(String[] args) throws Exception{
    System.out.println("Server is started");
    ServerSocket serverSocket = new ServerSocket(8090);
    System.out.println("Server is waiting for client");
    Socket socket = serverSocket.accept();
    System.out.println("Client connected");
    BufferedReader br = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
    String str = br.readLine();
    System.out.println("Client Data : "+str);
}
}
```

Output:



The screenshot shows a Java IDE's Run console with two tabs: 'SocketServer' and 'SocketClient'. The 'SocketServer' tab is active, displaying the following output:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home,  
Server is started  
Server is waiting for client  
Client connected  
Client Data : Md. Shoriful Islam Sakib  
  
Process finished with exit code 0
```