

Assignment no. 01 (Unit 01)

- 1) To create the server application, we need to create the instance of ServerSocket class. Here we are using the port number 6666 for the communication between the client and server. You may also choose any other port number. The accept() method waits for the client. If client connects with the given port number, it returns an instance of Socket.

→ **filename: Client.java**

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

public class Client {
    public static void main(String[] args) throws Exception {
        Socket s = new Socket("LocalHost", 6666);
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());
        dos.writeUTF("Connection has been established successfully.");
        System.out.println("Connected Successfully");
    }
}
```

filename: SocketServer.java

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

public class SocketServer {
    public static void main(String[] args) throws Exception {
        ServerSocket serverSocket = new ServerSocket(6666);
        Socket s = serverSocket.accept();
        DataInputStream dis = new DataInputStream(s.getInputStream());
        String str = (String) dis.readUTF();
        System.out.println("Client: " + str);
    }
}
```

- 2) Client will write first to the Server the Server will receive and print the text. Then server will write to the client and client will receive and print the text.

filename: Client.java

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

```
public class Q2_Client {
    public static void main (String [] args) throws Exception {
        Socket s = new Socket ("localhost", 1234);
        DataOutputStream dos = new DataOutputStream (s.getOutputStream ());
        InputStreamReader isr = new InputStreamReader (s.getInputStream ());
        BufferedReader br = new BufferedReader (isr);
        BufferedWriter readFromKeyboard = new BufferedWriter (new
InputStreamReader (System.in));
        String s1, s2;
        System.out.print (> );
        s1 = readFromKeyboard.readLine ();
        dos.writeByte (s1 + "\n");
        s2 = br.readLine ();
        System.out.println ("Server Server: " + s2);
    }
}
```

3

filename: Q2_Server.java

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

public class Q2_Server {
    public static void main (String [] args) throws Exception {
        ServerSocket serverSocket = new ServerSocket (1234);
        Socket s = serverSocket.accept();
        System.out.println("Connection successful");
        PrintStream ps = new PrintStream (s.getOutputStream());
        InputStreamReader isr = new InputStreamReader (s.getInputStream());
        BufferedReader br = new BufferedReader (isr);
        InputStreamReader isr2 = new InputStreamReader (System.in);
        BufferedReader br2 = new BufferedReader (isr2);

        String str1, str2;
        str1 = br.readLine();
        System.out.println ("client: " + str1);
        System.out.println (">");
        str2 = br.readLine();
        ps.println (str2);
    }
}
```

ps.close();

br.close();

br2.close();

ServerSocket.close();

s.close();

3 (PEER) takes care own connection destroying destination?

3 (Client) takes care own connection destroying destination?

("I assume nothing") nothing to do

3. The `URLConnection` class provides many methods. We can display all the data of a webpage by using the `getInputStream()` method. It returns all the data of the specified URL in the stream that can be read and displayed.

filename: `URLConnectionClass.java`

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

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

```
public class URLConnectionClass {
```

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

```
        URL url = new URL("http://www.pratyayphond.me/Self-Assess/");
```

```
        about-us.html");
```

```
        URLConnection urlConnection = url.openConnection();
```

```
        InputStream in = urlConnection.getInputStream();
```

```
        int i;
```

```
        while ((i = in.read()) != -1) {
```

```
            System.out.print((char) i);
```

```
}
```

System.out.println();
System.out.println("Host :" + url.getHost());
System.out.println("Protocol :" + url.getProtocol());
System.out.println("File :" + url.getFile());

3

4) Simple example of InetAddress class to get IP Address of www.javatpoint.com website.

```
import java.io.*;  
import java.net.*;  
  
public class Q4 {  
    public static void main(String[] args) throws Exception {  
        InetAddress inet = InetAddress.getByName("www.javatpoint.com");  
        System.out.println("Host name :" + inet.getHostName());  
        System.out.println("IP :" + inet.getIP(getHostAddress()));  
    }  
}
```

5) Sending and receiving datagram packets by Datagram Socket.

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

filename : Q5-client.java

```

import java.net.*;           // (1) InBring, fun, notes
import java.util.Scanner;    // (2) (localHost + " : 7070") InBring, fun, notes
                            // ((LocalHost + " : 7070") InBring, fun, notes)
public class Q5_Client {
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();
        String str = null;
        Scanner sc = new Scanner(System.in);
        byte[] b = null;
        InetAddress inet = InetAddress.getLocalHost();
        str = sc.nextLine();
        b = str.getBytes();
        DatagramPacket dp = new DatagramPacket(b, b.length, inet, 2003);
        ds.send(dp);
    }
}

import java.io.*;           // (3) InBring, fun, notes
import java.net.*;           // (4) (localHost + " : 7070") InBring, fun, notes
                            // ((LocalHost + " : 7070") InBring, fun, notes)
public class Q5_Server {
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(2003);
        byte[] b = new byte[1024];
    }
}

```

```
DatagramPacket dp = new DatagramPacket(b, b.length);  
ds.receive(dp);
```

```
String str = new String(dp.getData());
```

```
System.out.println("Connection Established");
```

```
System.out.println("Client: " + str.trim());
```

}

}