

## Quiz #8

Java Programming II

May 22, 2018 (6-7 period)

Class: C6 ID Number: s1240234 Name: Yuta Nemoto

Fill out the blanks with suitable words.

1. In network protocol, especial in datagram sockets and packets,  
(Transmission Control Protocol) provides reliable and sequenced data exchange.  
(User Datagram Protocol) provides a more efficient, best-effort delivery, but less reliable.

2. Write proper statements in the blank of the program below.

// The programs run in the same host.

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

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

```
import java.util.*;
```

```
class ServerSocketDemo {
```

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

```
        try {
```

```
            // Get Port
```

```
            int port = 4321;
```

```
            Random random = new Random();
```

```
            //Create Server Socket
```

```
            ServerSocket svrskt = new ServerSocket(port);
```

```
            //Create Infinite Loop
```

```
            while(true) {
```

```
                //Accept Incoming Requests by ServerSocket
```

```
                (Socket skt = svrskt.accept());
```

```
                //Write Result to Client
```

```

        OutputStream os = skt.getOutputStream();
        DataOutputStream dos = new DataOutputStream(os);
        dos.writeInt(random.nextInt());

        //Close socket
        skt.close();
    }
}
catch (Exception e) {
    System.out.println("Exception: " + e);
}
}
}

```

(Run) % java ServerSocketDemo

```

class SocketDemo {
    public static void main(String args[]) {
        try {
            //Get Server and Port
            String server = "localhost";
            int port = Integer.parseInt(args[0]);
            //Create socket
            Socket clientskt = new Socket(server, port);
            // Get InputStream from the Socket object
            InputStream is = (clientskt.getInputStream());
            DataInputStream dis = new DataInputStream(is);
            int i = dis.readInt();
            //Display Result
            System.out.println(i);
            //Close Socket
            s.close();
        }
        catch (Exception e) {
            System.out.println("Exception: " + e);
        }
    }
}

```

(Run) % java SocketDemo 4321