

# The Internet

Nick Troccoli  
CSBridge CTU 2019

# Learning Goals

1. Write a program that can make internet requests
2. Write a program that can respond to internet requests



# Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo:* echo
- *Demo:* Chat

# Plan For Today

- **How do internet programs work?**
- Sending Data Over The Internet
- Writing a client and server
- *Demo:* echo
- *Demo:* Chat

# Programs and the Internet

How does your  
phone  
communicate with  
Facebook?

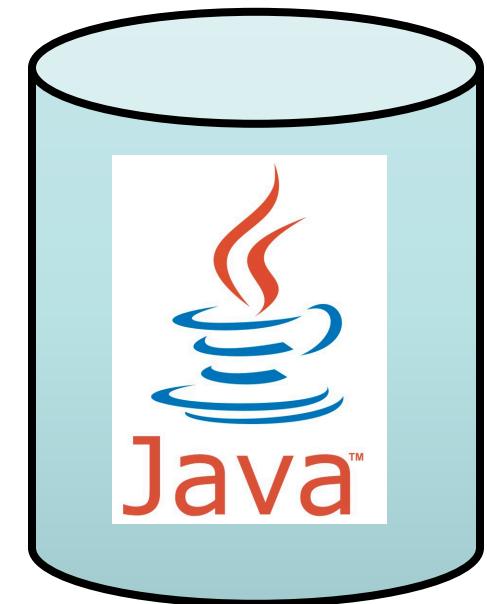
# Programs and the Internet

The Java program  
on your phone talks  
to the Java program  
at Facebook.

# Facebook

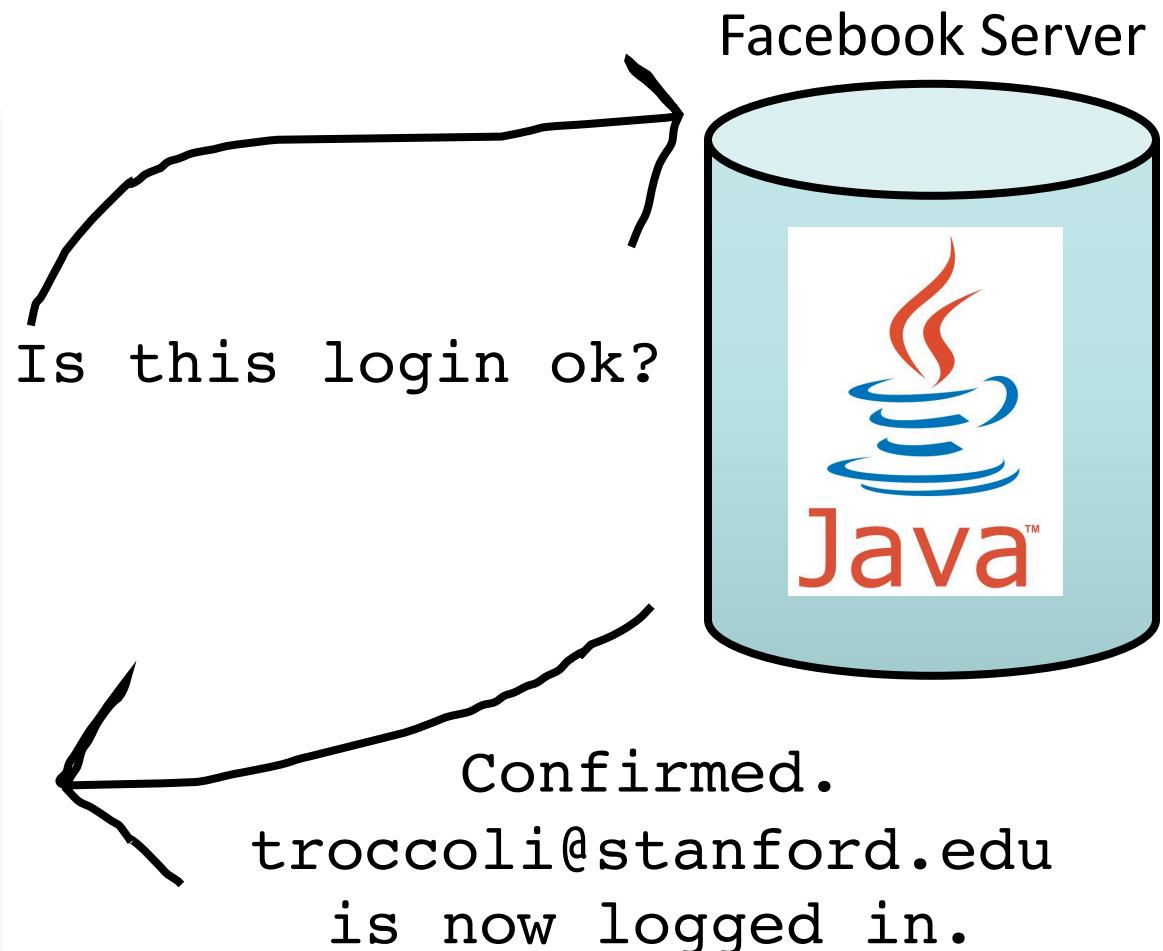
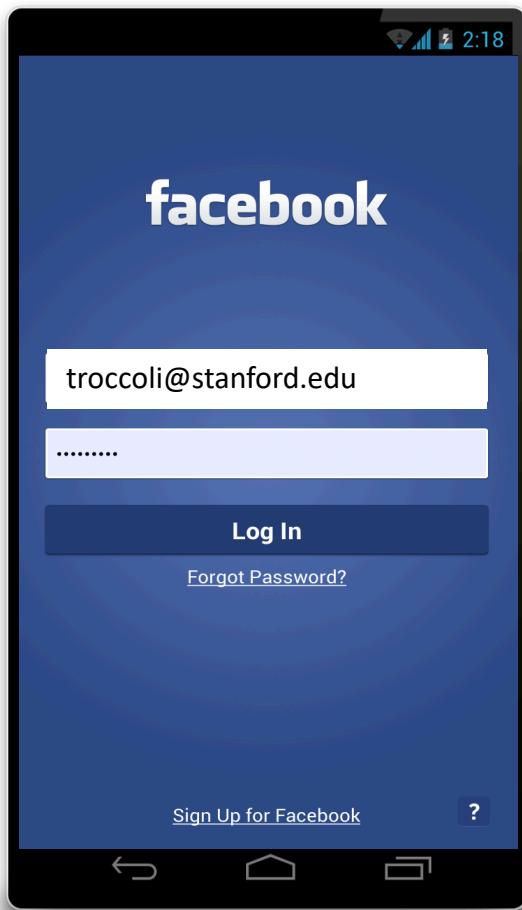


Facebook Server

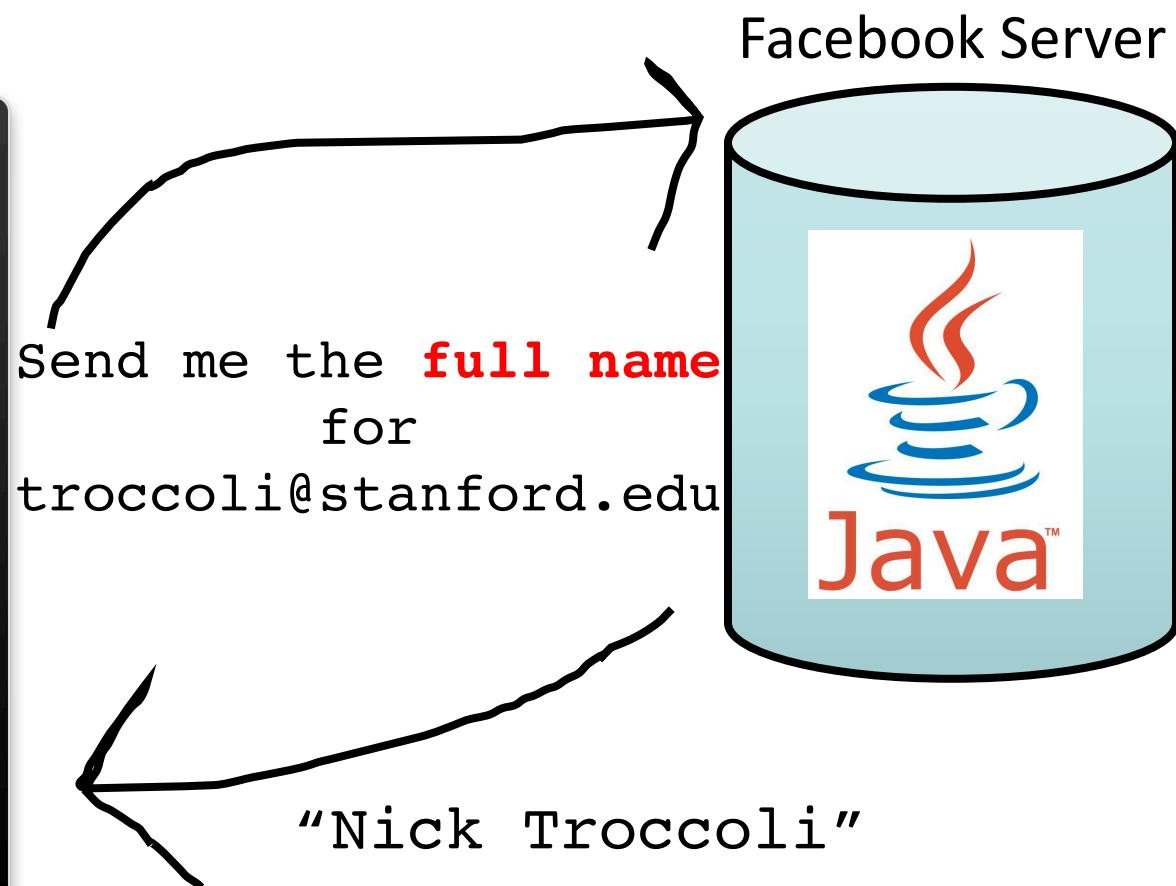
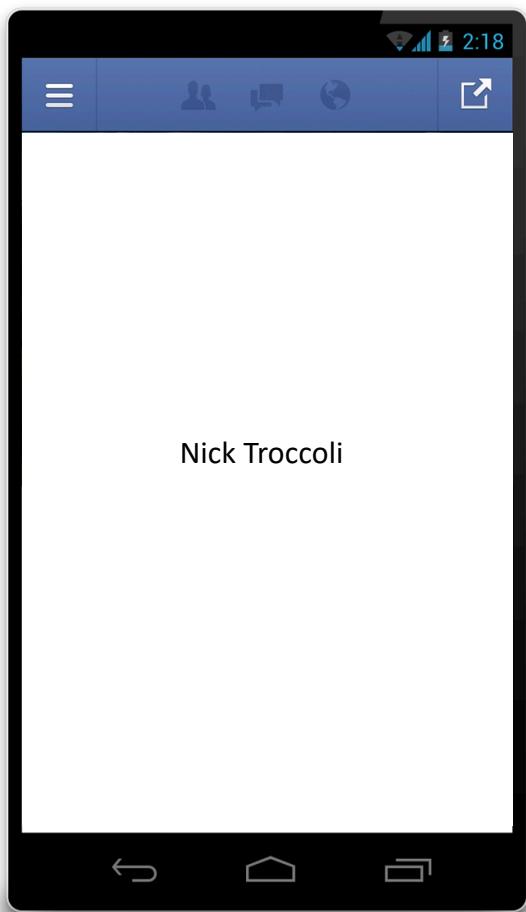


\* Android phones run Java. So do Facebook servers!

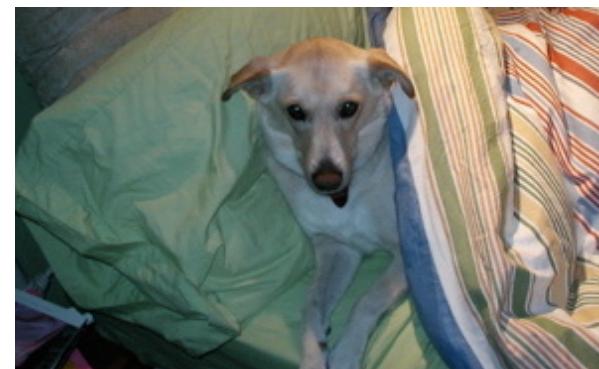
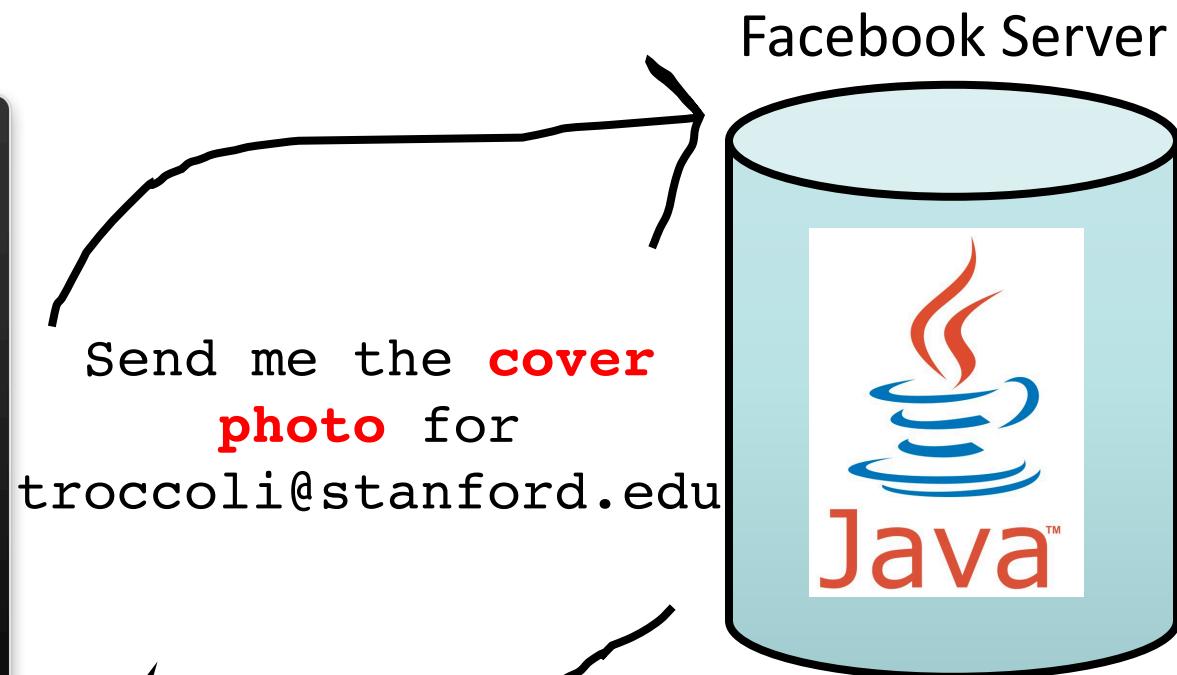
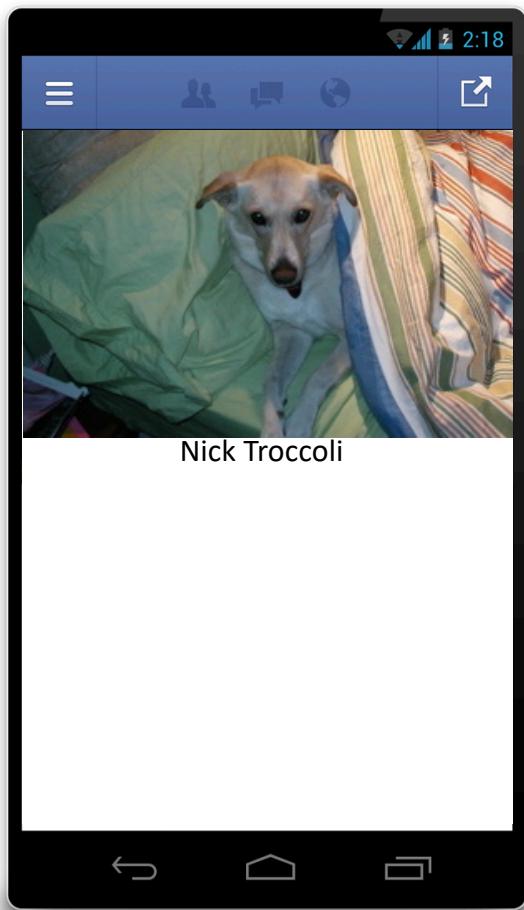
# Facebook



# Facebook



# Facebook



# Programs and the Internet

There are two types  
of internet  
programs: clients  
and servers.

# Programs and the Internet

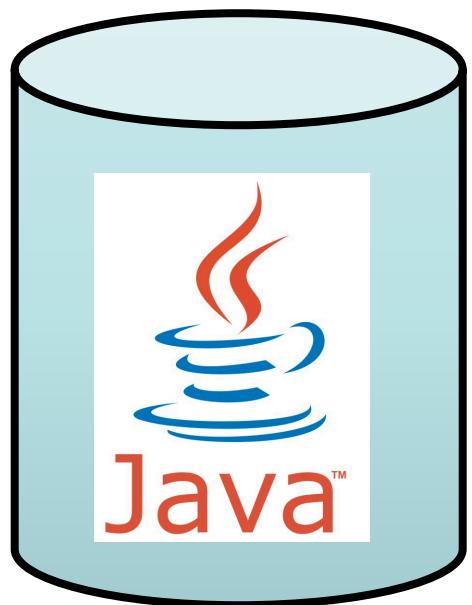
Servers are  
constantly running,  
listening for requests  
to respond to.

# Programs and the Internet

Clients send requests  
to servers for  
information they need.

# Servers Are Computer Programs!

Facebook Server



=



# Background: The Internet



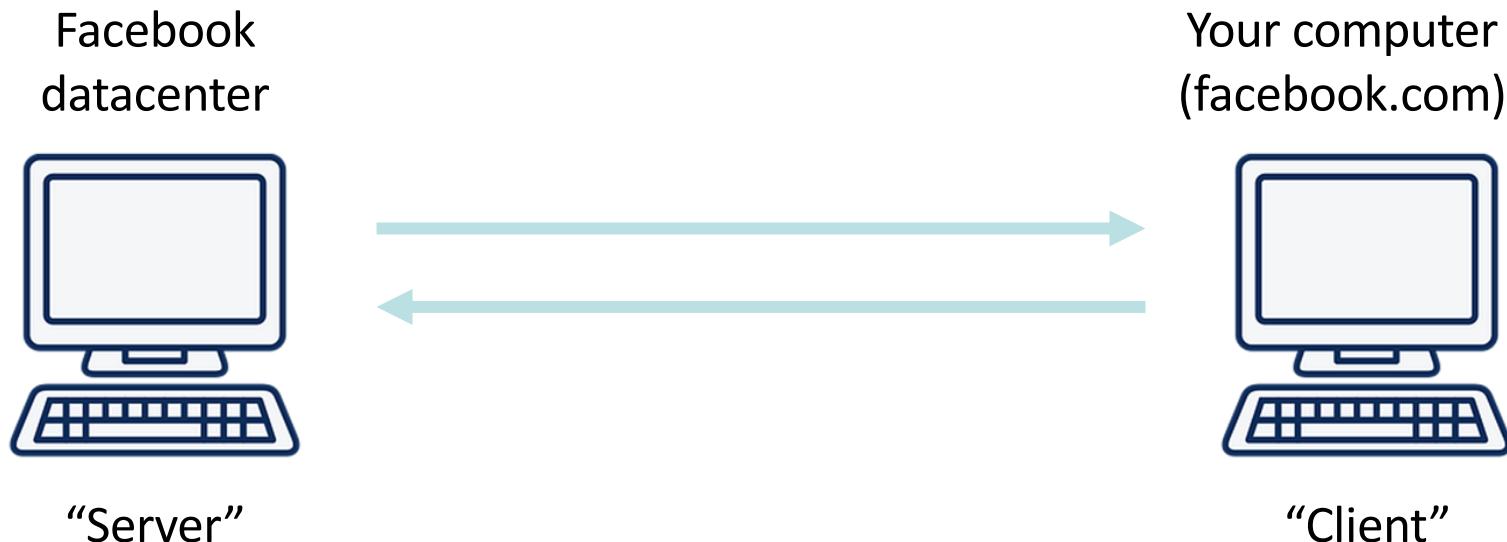
The internet is just many programs sending messages (as *Strings*)

# Background: The Internet



The internet is just many programs sending messages (as *Strings*)

# Background: The Internet



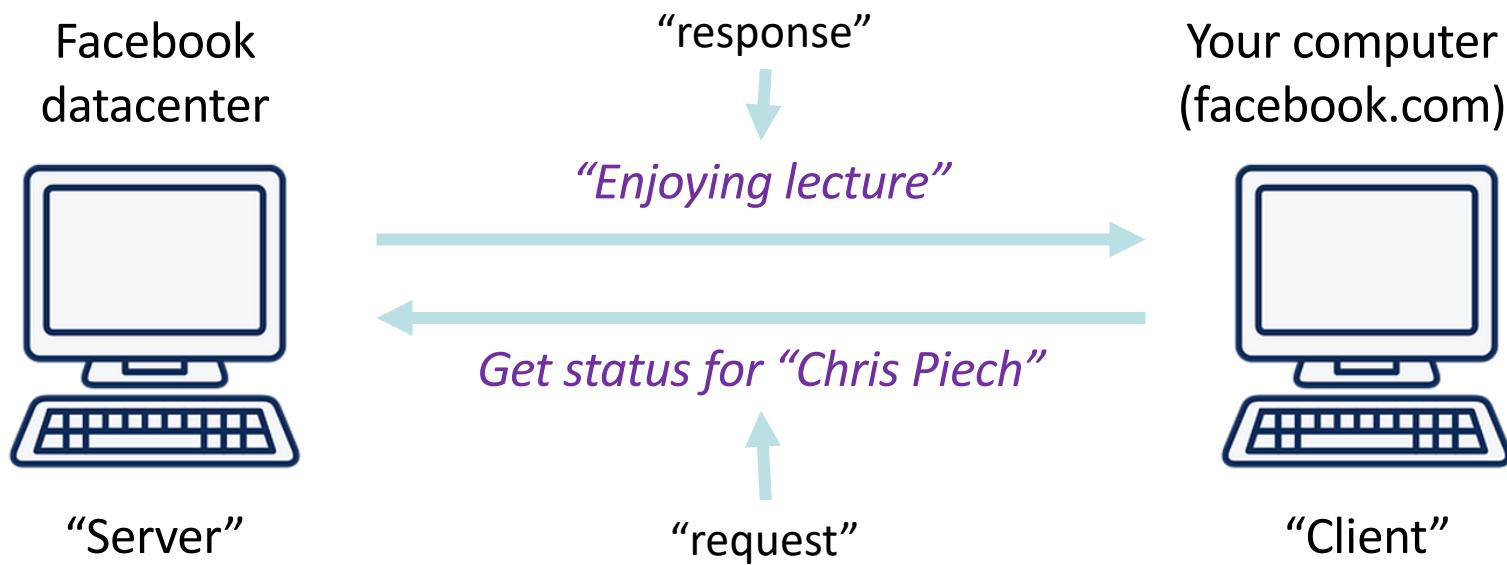
The internet is just many programs sending messages (as *Strings*)

# Background: The Internet



The internet is just many programs sending messages (as *Strings*)

# Background: The Internet



The internet is just many programs sending messages (as *Strings*)



## The Internet

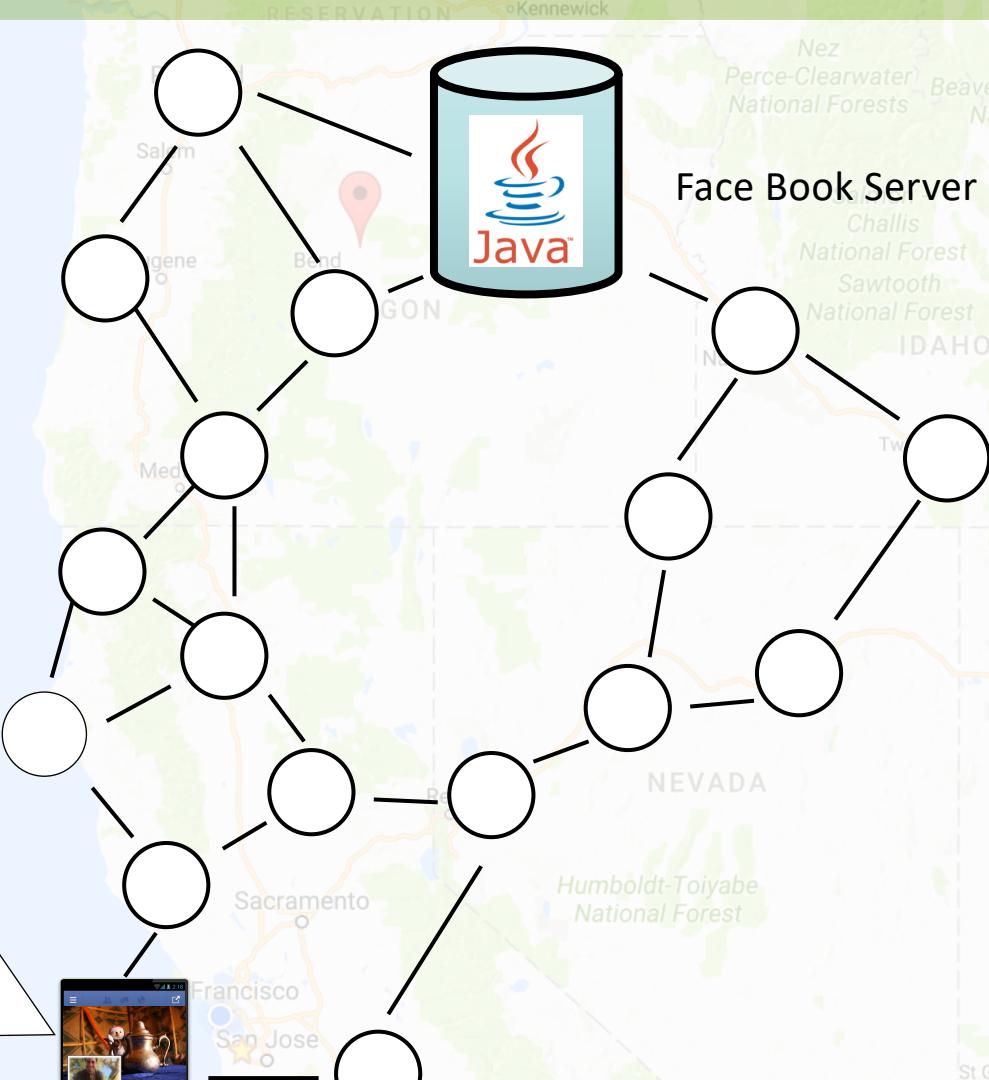


Earth

Google

The Internet

Get status for  
piech@cs.stanford.edu

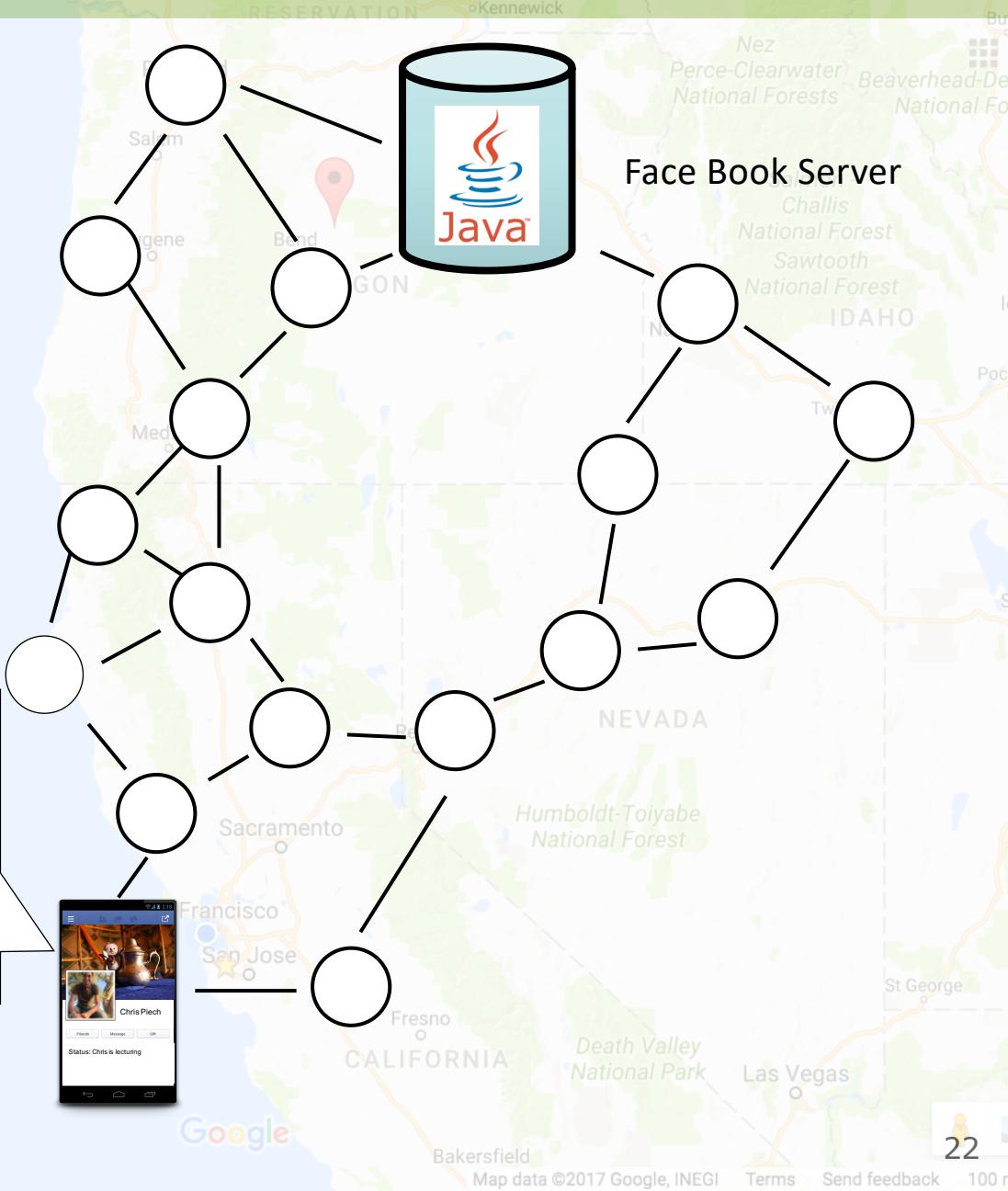
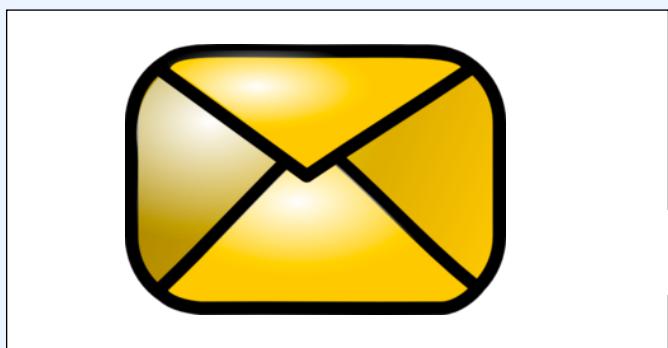




Secure

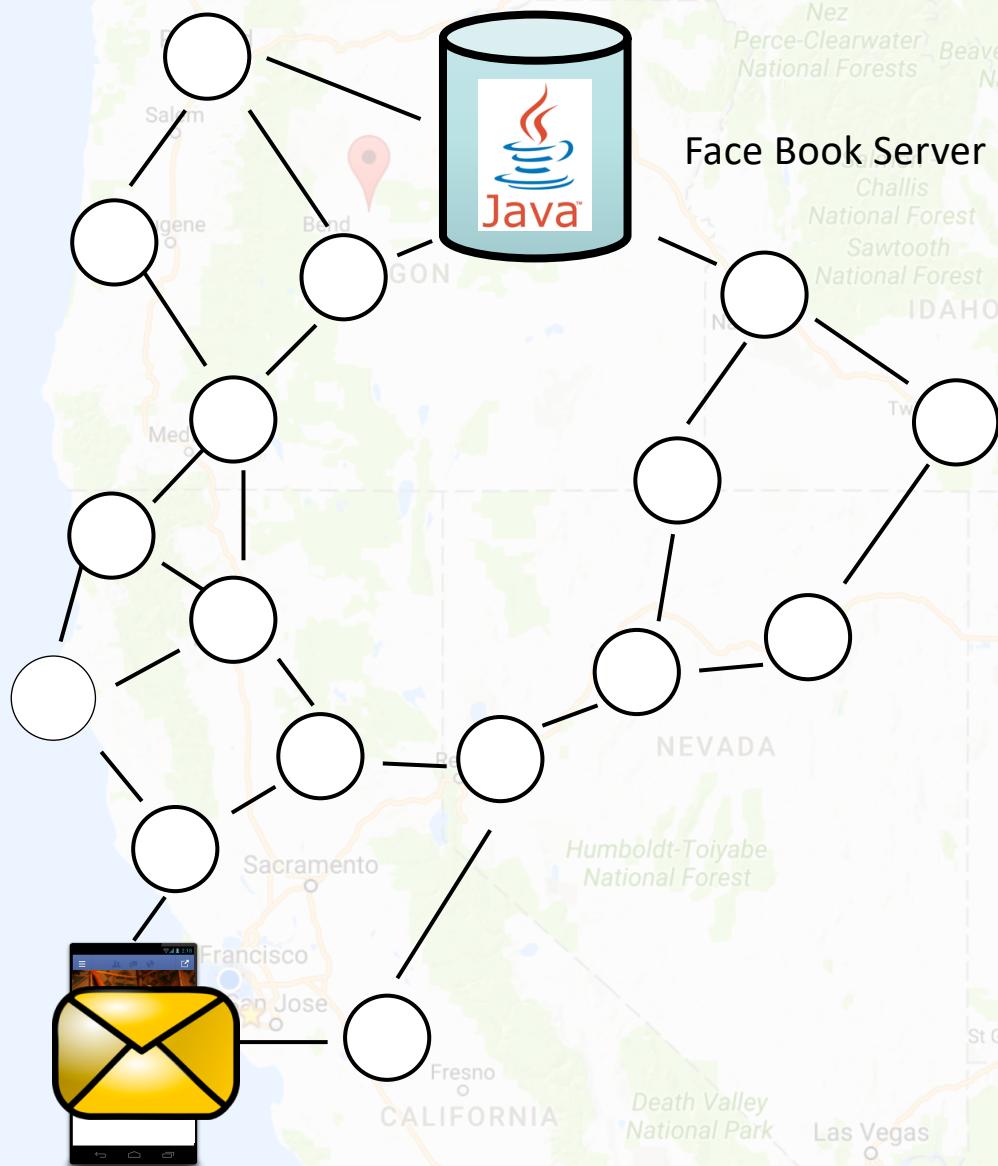
<https://www.google.com/maps/place/Prineville,+OR+97754/@40.9703226,-122.0642667,6z/data=!4m5!3m4!1s...>

## The Internet



Earth

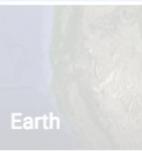
## The Internet



Earth

teaching

The Internet



Earth

Google



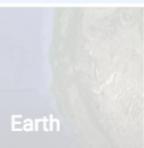
## The Internet



Earth



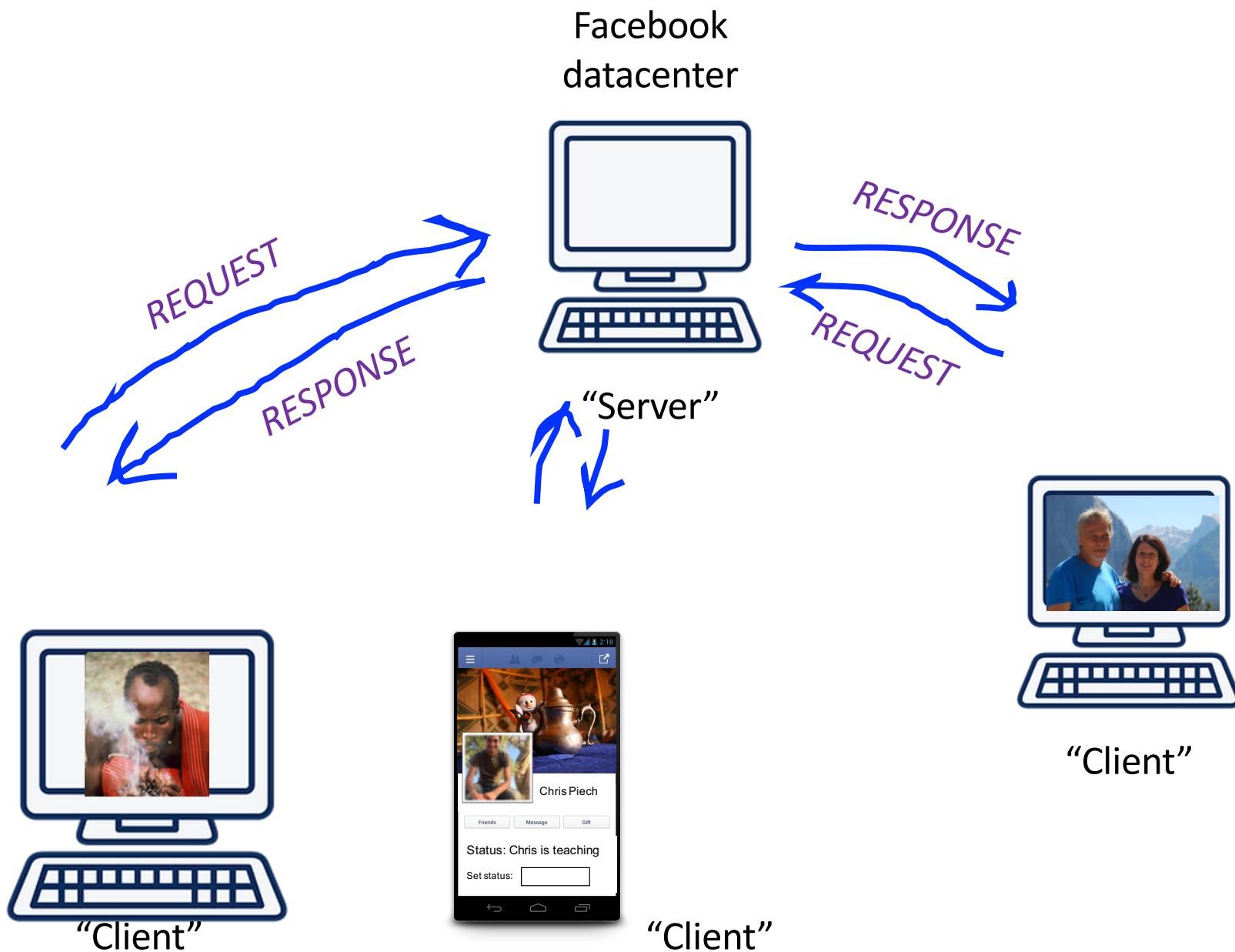
## The Internet



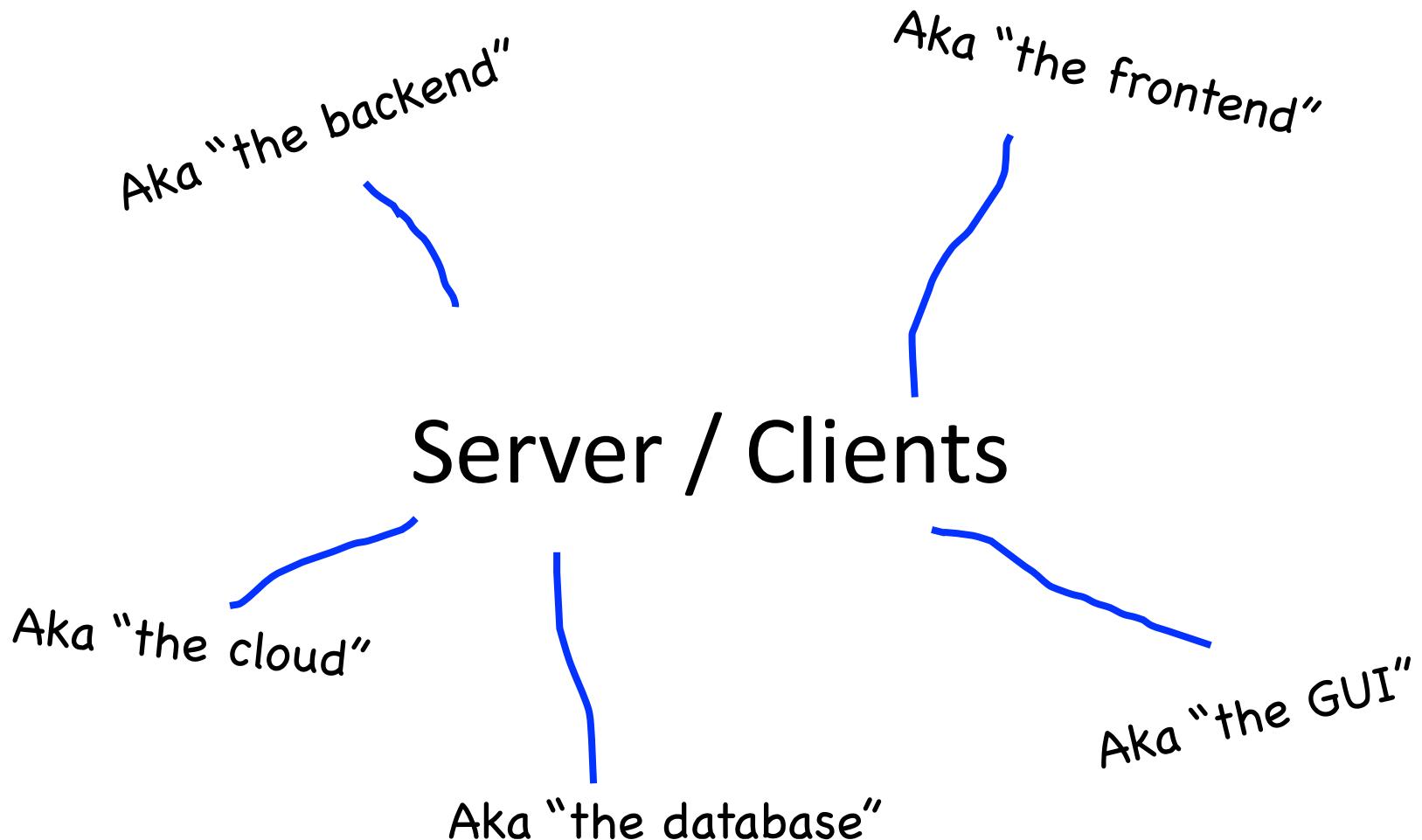
Earth

Many programs can connect  
to the same server

# The Internet



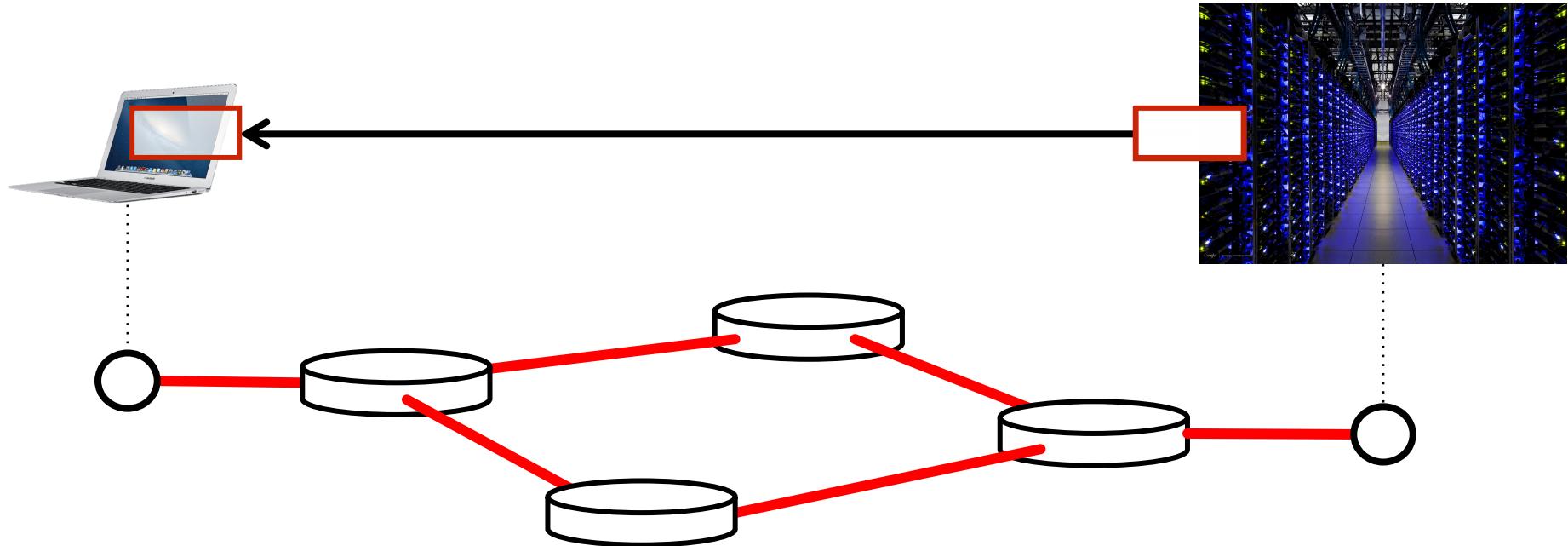
# Most of the Internet



# Plan For Today

- How do internet programs work?
- **Sending Data Over The Internet**
- Writing a client and server
- *Demo:* echo
- *Demo:* Chat

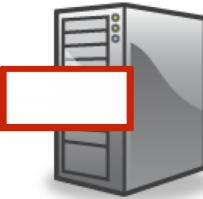
Google YouTube facebook

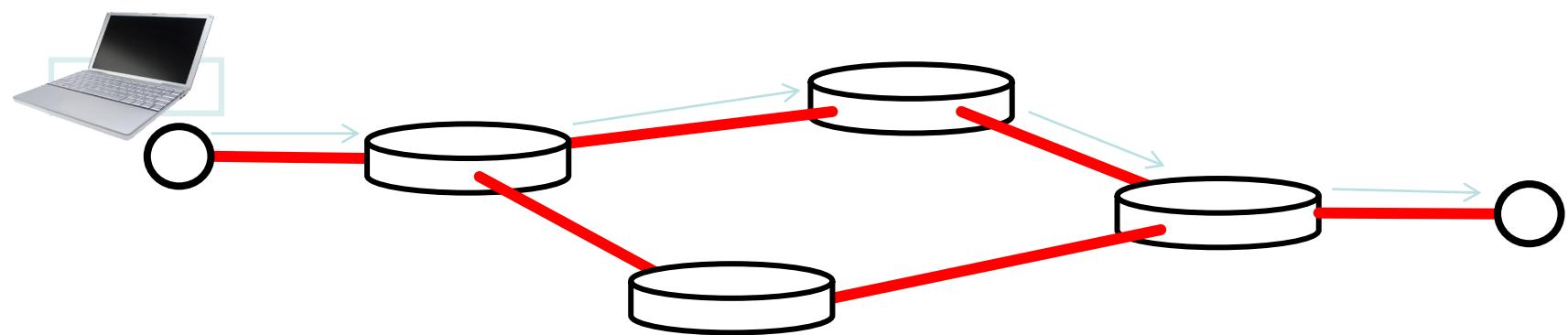


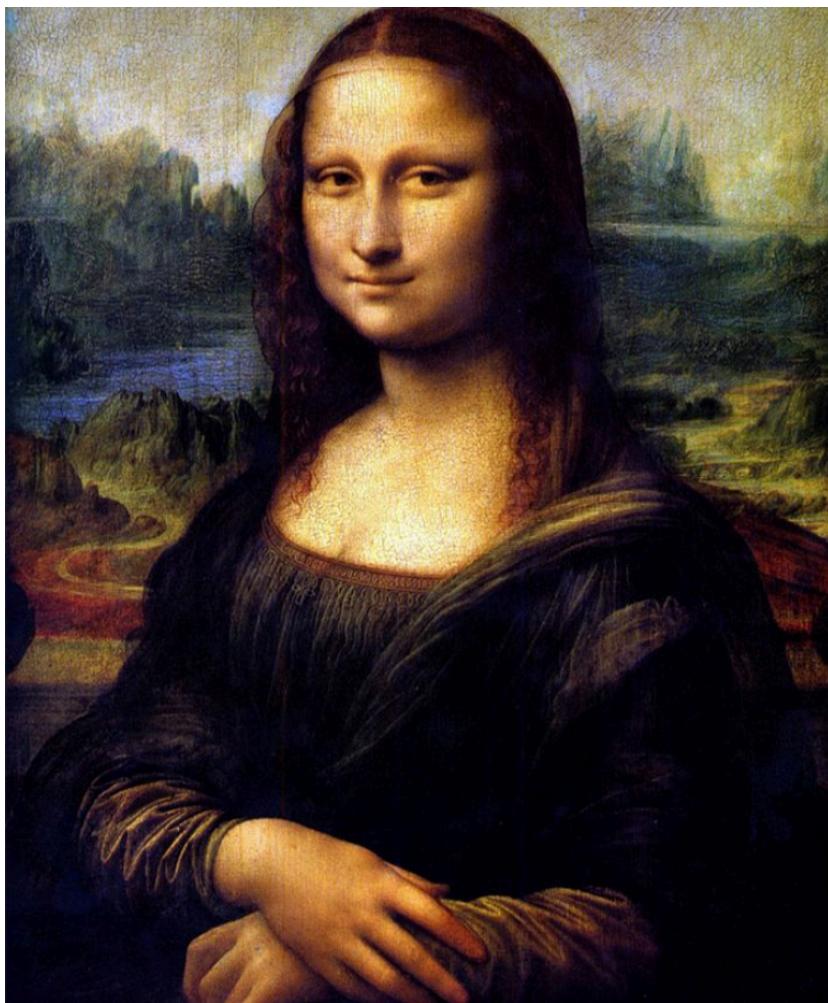
My Java Program

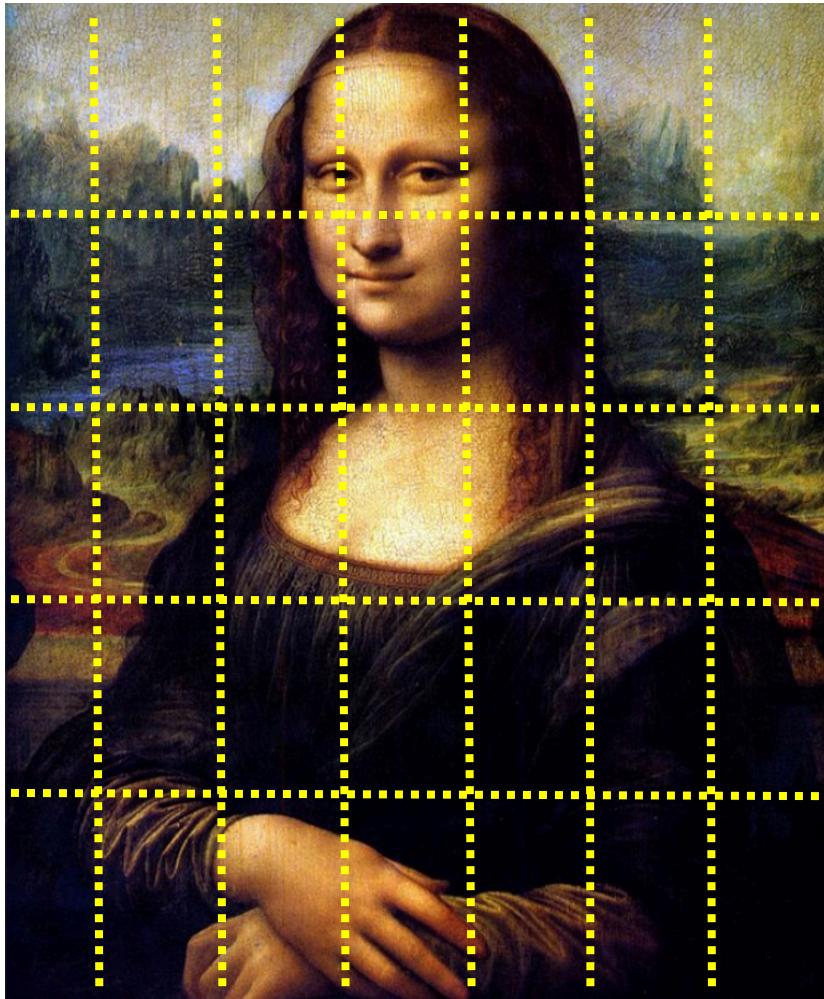


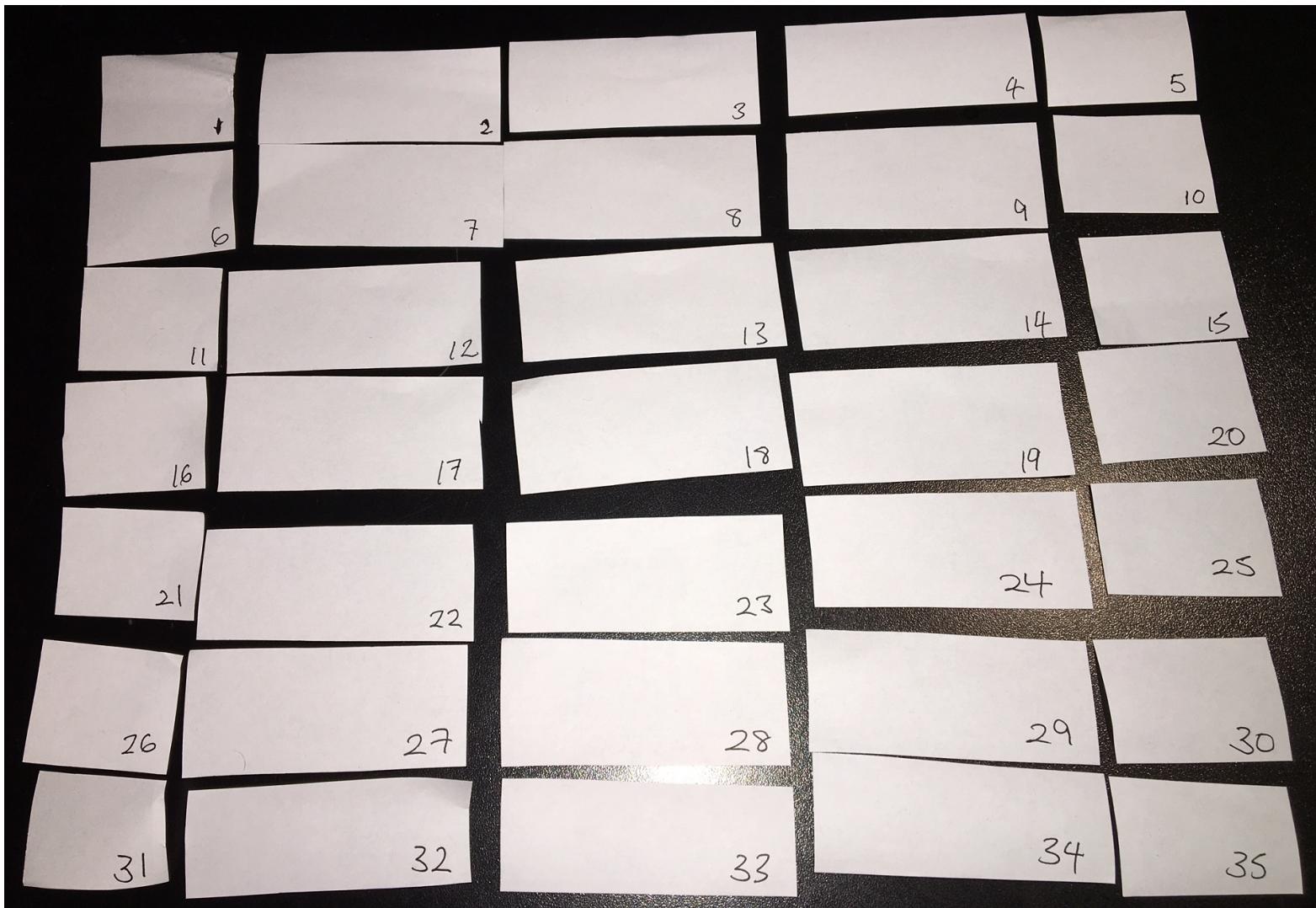
Someone else's Java Program



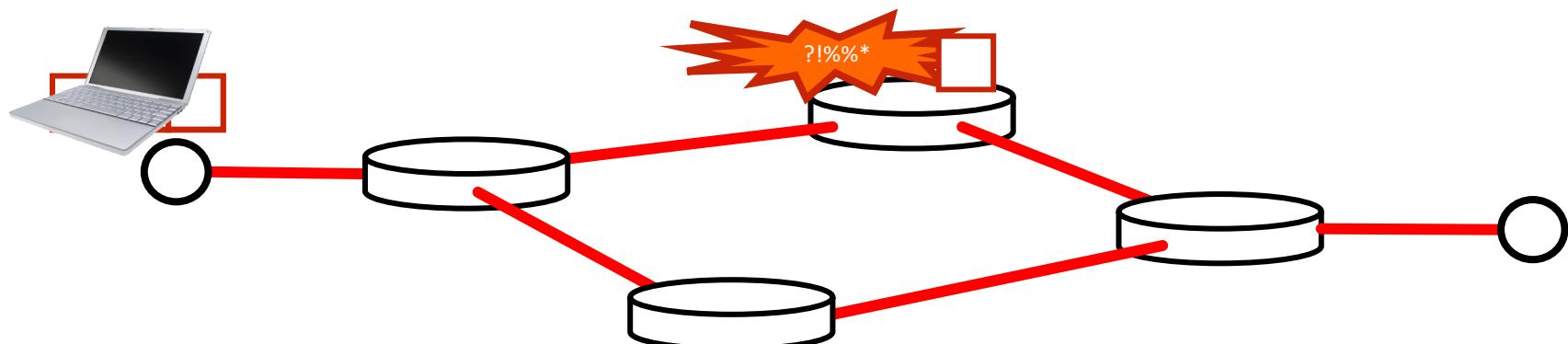




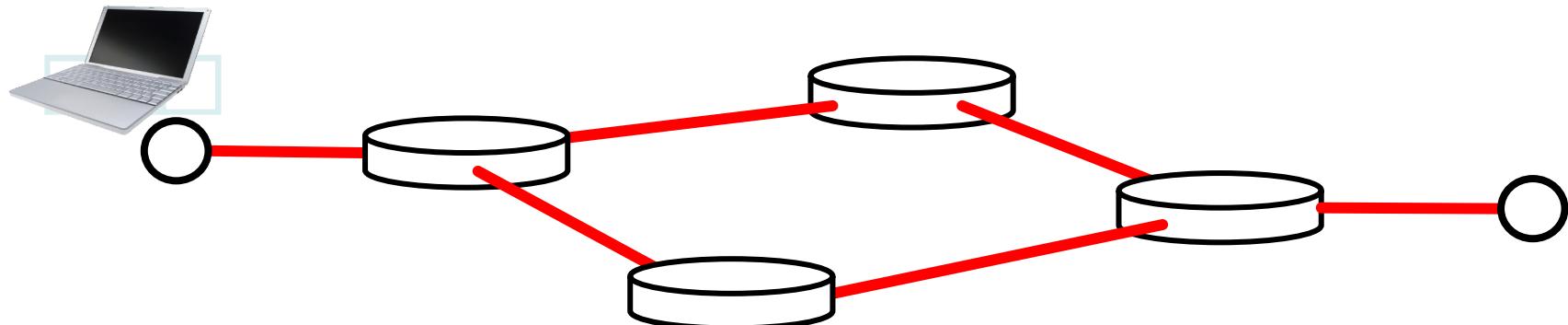




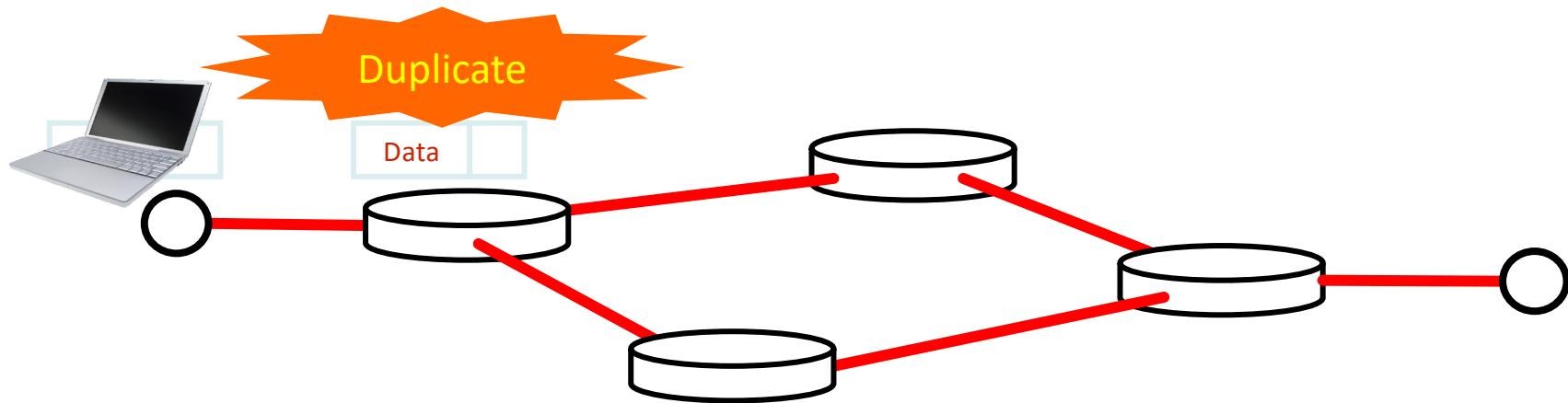
# Packets may be damaged



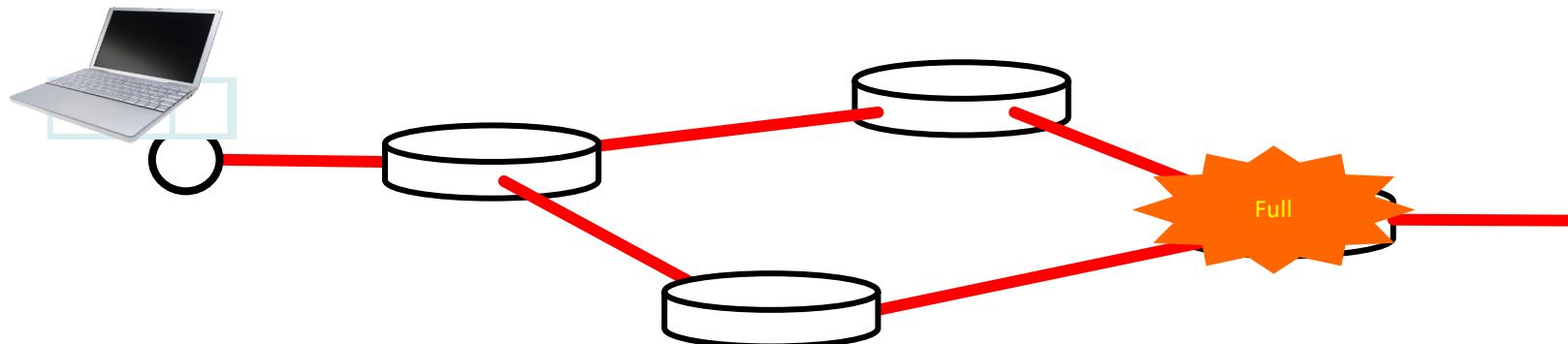
# Packets may arrive out of order



# Packets may be duplicated

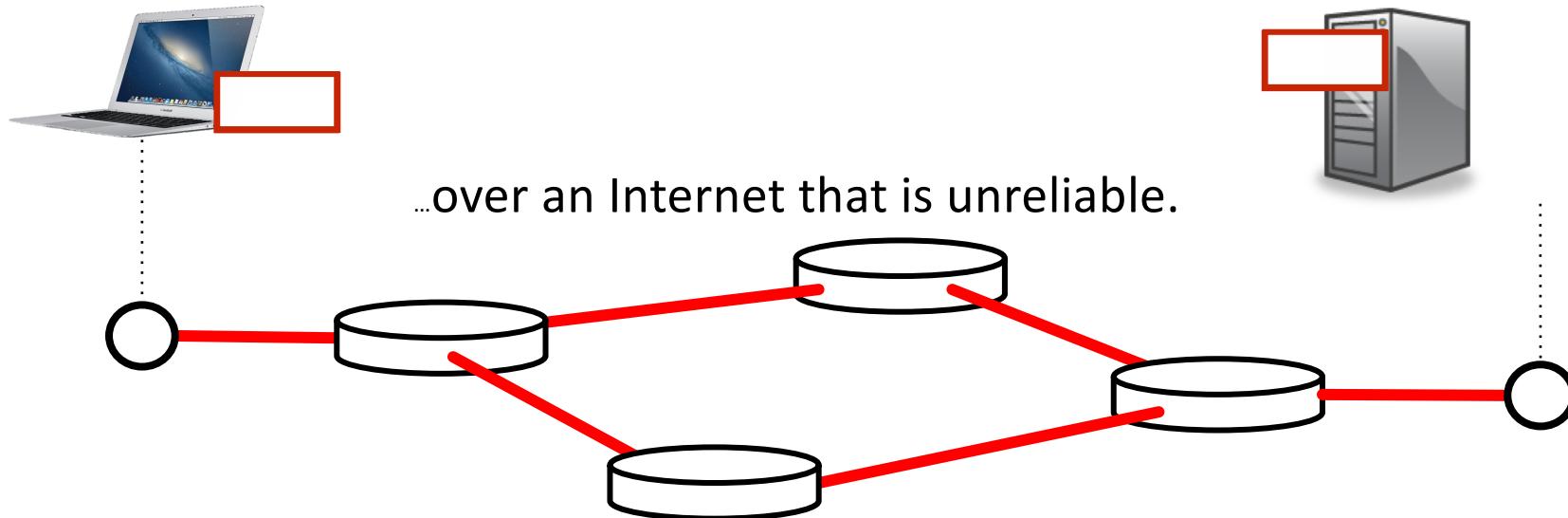


# They may not arrive at all!



# The Internet Is Unreliable

Applications send and receive data in packets....



Software on the devices themselves compensates for this.

# **Let's send a message over the “CSBridgeNet”**

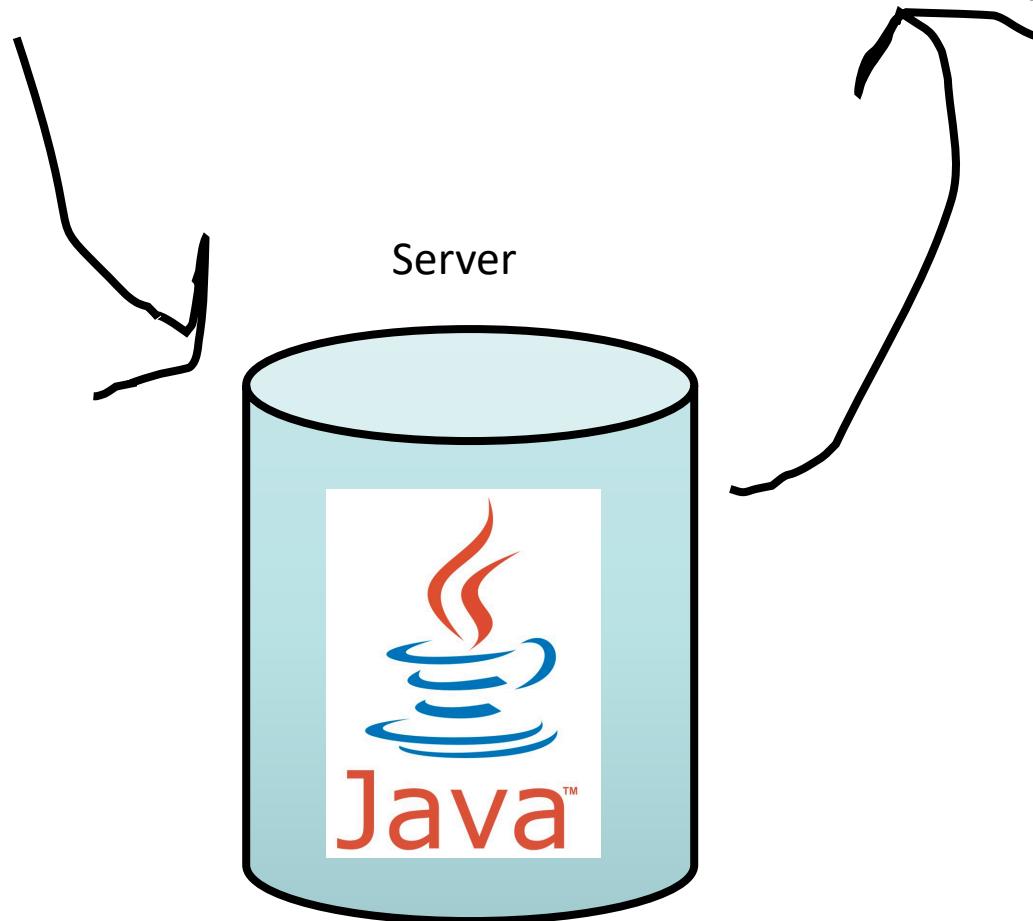
# Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- **Writing a client and server**
- *Demo:* echo
- *Demo:* Chat

# Servers

**Request**  
someRequest

**String**  
serverResponse



# Servers on one slide

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

2

```
// make a Server object  
private SimpleServer server  
= new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

# A Server's Simple Purpose

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

2

```
// make a Server object  
private SimpleServer server  
    = new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

# A Server's Simple Purpose

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

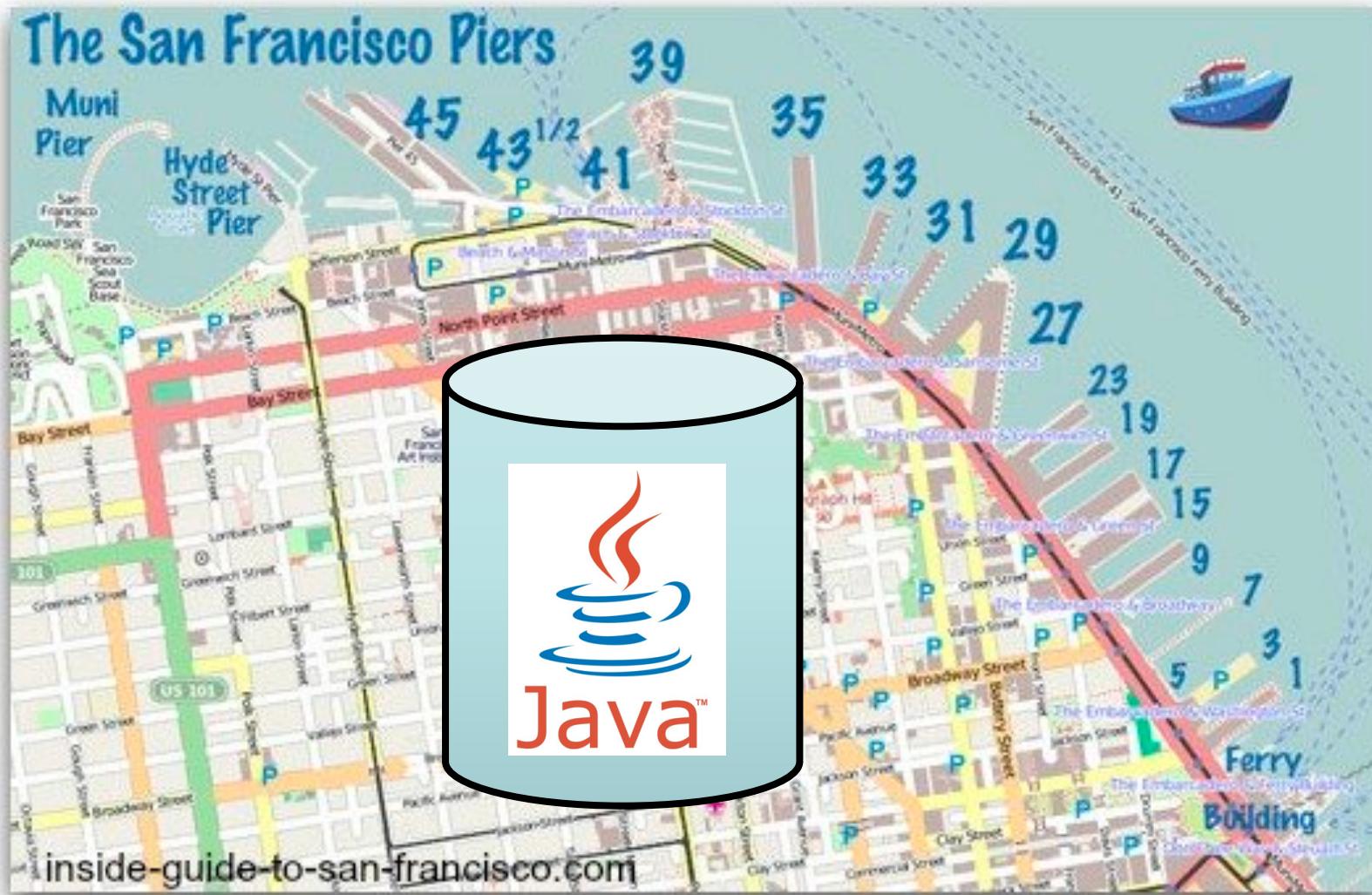
2

```
// make a Server object  
private SimpleServer server  
= new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

# What is a Port?



# Servers on one slide

1

```
public String requestMade(Request request) {  
    // server code goes here  
}
```

2

```
// make a Server object  
private SimpleServer server  
= new SimpleServer(this, 8000);
```

3

```
public void run(){  
    // start the server  
    server.start();  
}
```

# What is a Request?



```
/* Request has a command */  
String command;  
  
/* Request has parameters */  
HashMap<String, String> params;
```

## Request request

---

```
// methods that the server calls on requests  
request.getCommand();  
request.getParam(key); //returns associated value
```

# Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- ***Demo: echo***
- *Demo: Chat*

# Our First Server

```
public class EchoServer extends ConsoleProgram implements  
SimpleServerListener {  
  
    private SimpleServer server = new SimpleServer(this, 8090);  
  
    public void run() {  
        server.start();  
        println("Starting server...");  
    }  
  
    public String requestMade(Request request) {  
        String cmd = request.getCommand();  
        int cmdLength = cmd.length();  
        return "Your command was " + cmdLength + " chars long.";  
    }  
}
```

# Our First Server

```
public class EchoServer extends ConsoleProgram implements  
SimpleServerListener {  
  
    private SimpleServer server = new SimpleServer(this, 8090);  
  
    public void run() {  
        server.start();  
        println("Starting server...");  
    }  
  
    public String requestMade(Request request) {  
        String cmd = request.getCommand();  
        int cmdLength = cmd.length();  
        return "Your command was " + cmdLength + " chars long.";  
    }  
}
```

# Our First Server

```
public class EchoServer extends ConsoleProgram implements  
SimpleServerListener {  
  
    private SimpleServer server = new SimpleServer(this, 8090);  
  
    public void run() {  
        server.start();  
        println("Starting server...");  
    }  
  
    public String requestMade(Request request) {  
        String cmd = request.getCommand();  
        int cmdLength = cmd.length();  
        return "Your command was " + cmdLength + " chars long.";  
    }  
}
```

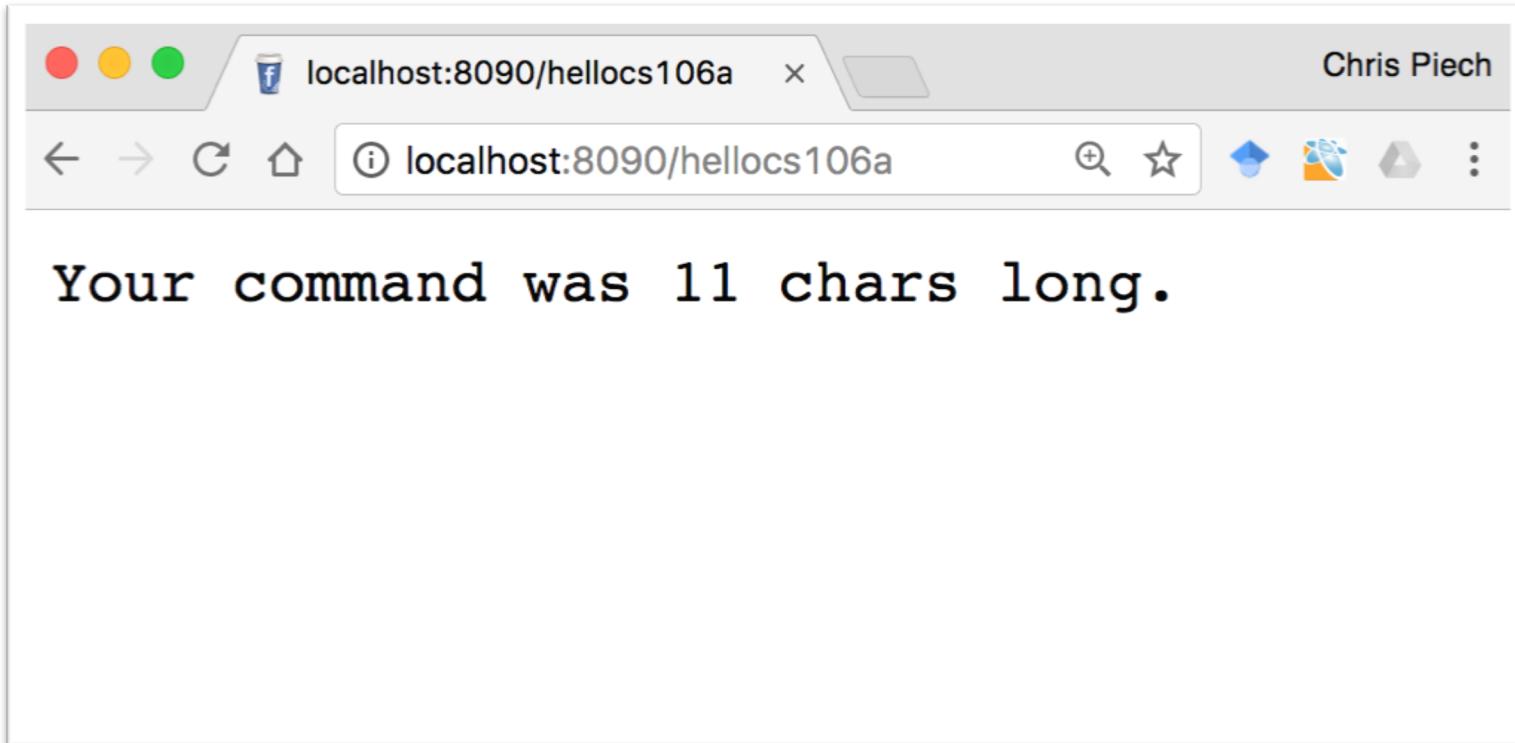
# Our First Server

```
public class EchoServer extends ConsoleProgram implements  
SimpleServerListener {  
  
    private SimpleServer server = new SimpleServer(this, 8090);  
  
    public void run() {  
        server.start();  
        println("Starting server...");  
    }  
  
    public String requestMade(Request request) {  
        String cmd = request.getCommand();  
        int cmdLength = cmd.length();  
        return "Your command was " + cmdLength + " chars long.";  
    }  
}
```

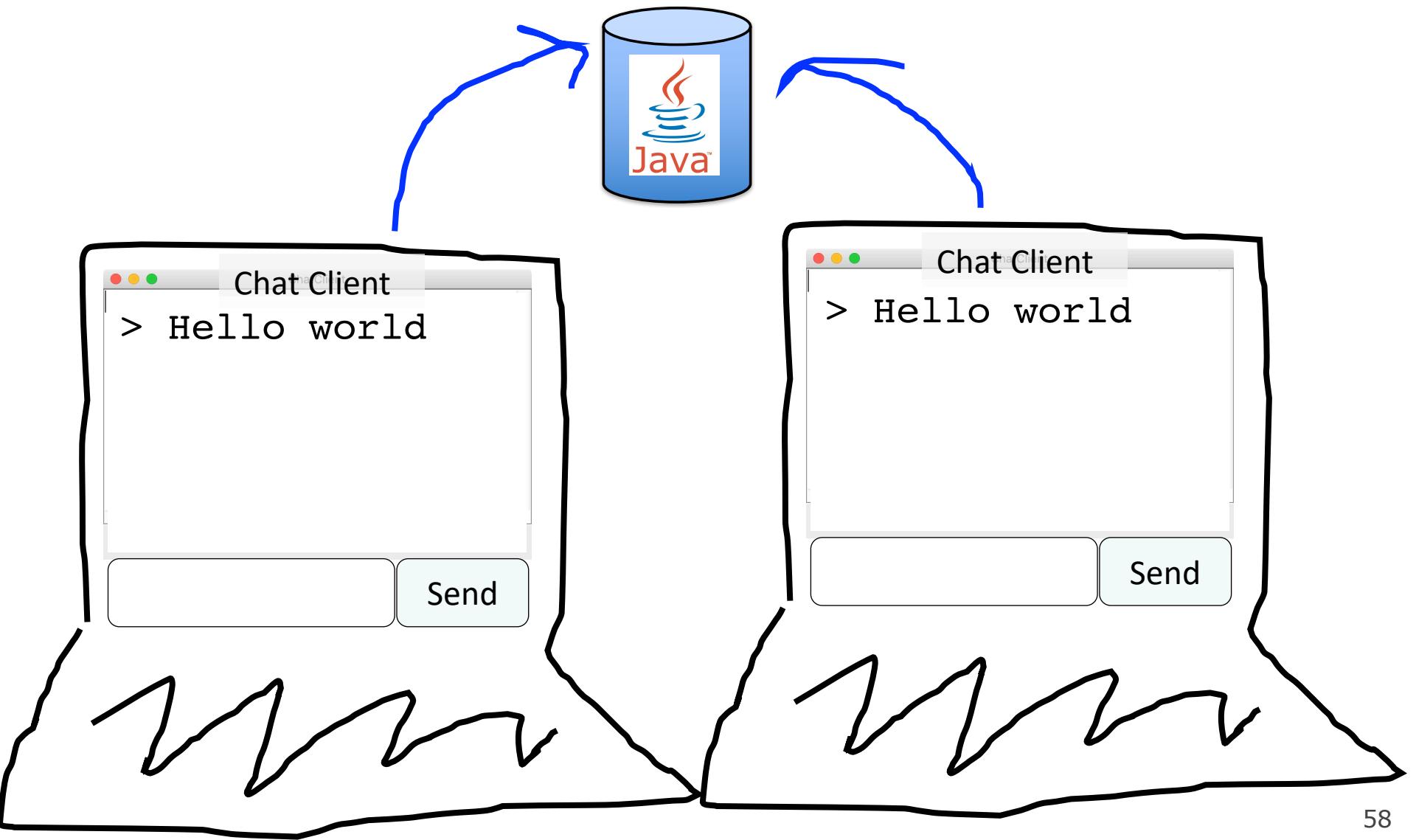
# Our First Server

```
public class EchoServer extends ConsoleProgram implements  
SimpleServerListener {  
  
    private SimpleServer server = new SimpleServer(this, 8090);  
  
    public void run() {  
        server.start();  
        println("Starting server...");  
    }  
  
    public String requestMade(Request request) {  
        String cmd = request.getCommand();  
        int cmdLength = cmd.length();  
        return "Your command was " + cmdLength + " chars long.";  
    }  
}
```

# Echo Server



# Clients



# Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

# Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

# Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

# Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
}
```

# Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
}
```

# Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

# Clients on one slide

```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

# Clients on one slide

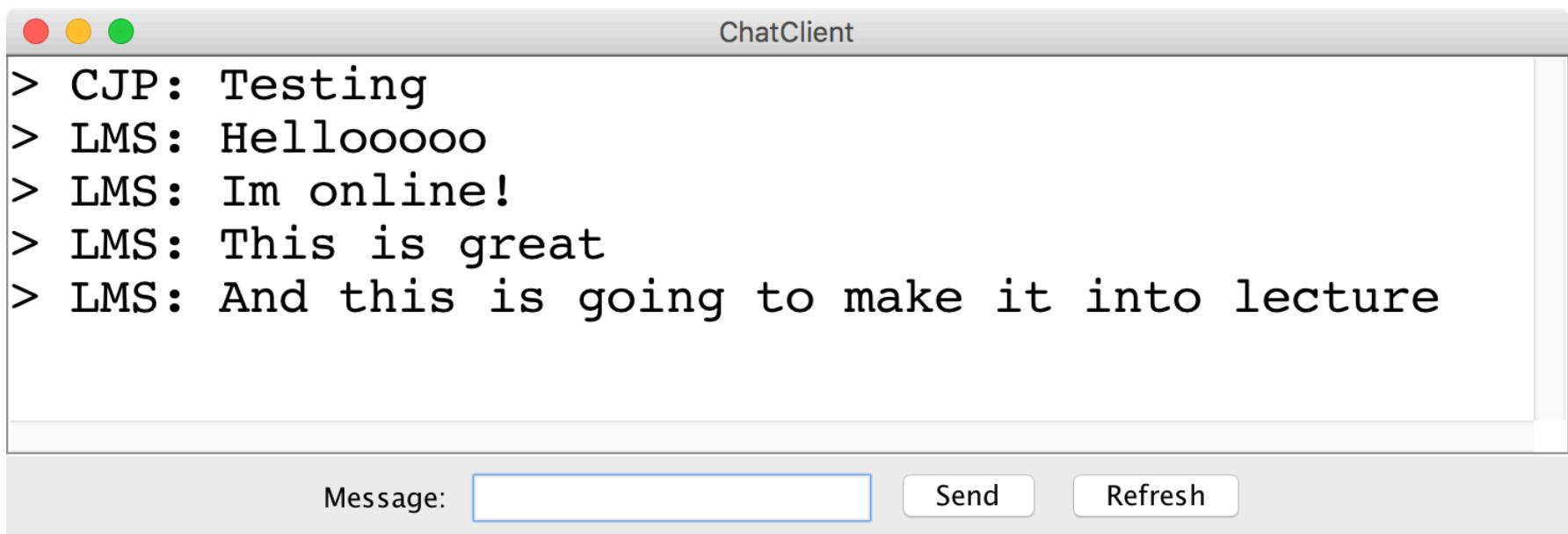
```
try {  
  
    // 1. construct a new request  
    Request example = new Request("getStatus");  
  
    // 2. add parameters to the request  
    example.addParam("name", "chris");  
  
    // 3. send the request to a computer on the internet  
    String result = SimpleClient.makeRequest(HOST, example);  
  
} catch(IOException e) {  
  
    // The internet is a fast and wild world my friend  
  
}
```

# **Example: Echo Client**

# Plan For Today

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo:* echo
- ***Demo: Chat***

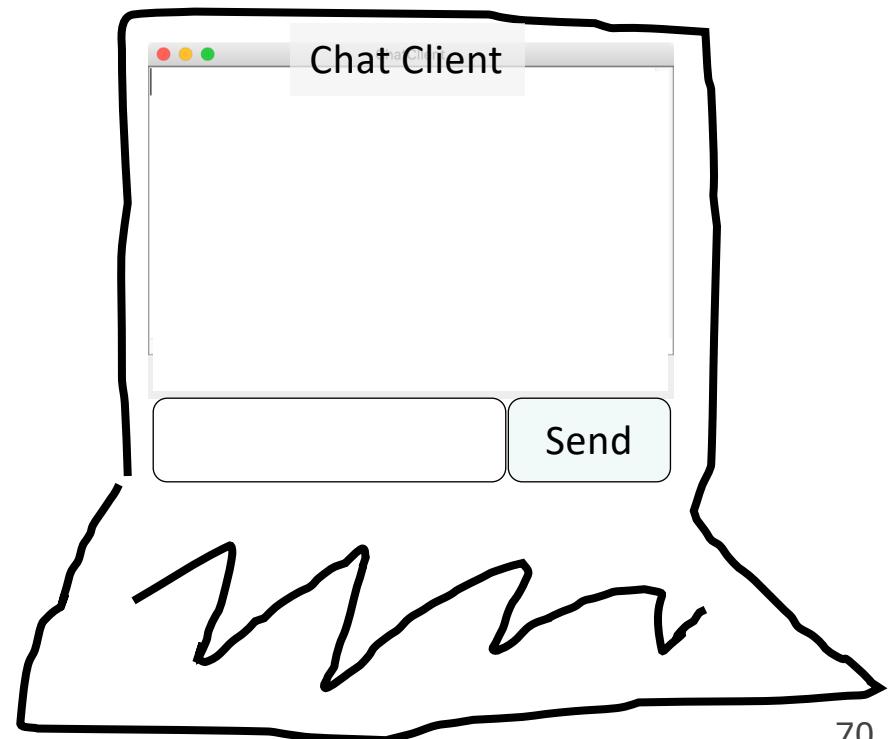
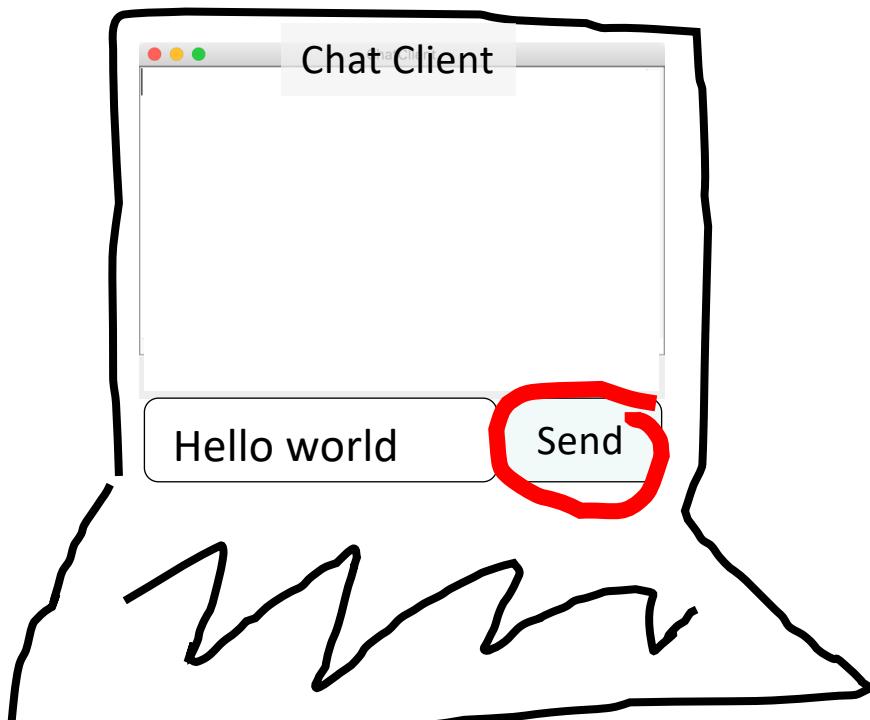
# Chat Server and Client

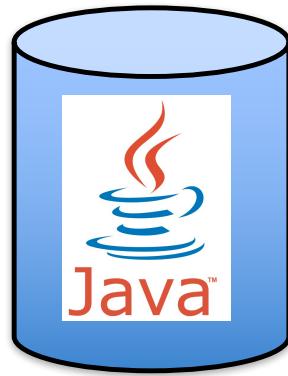




```
history = [  
]
```

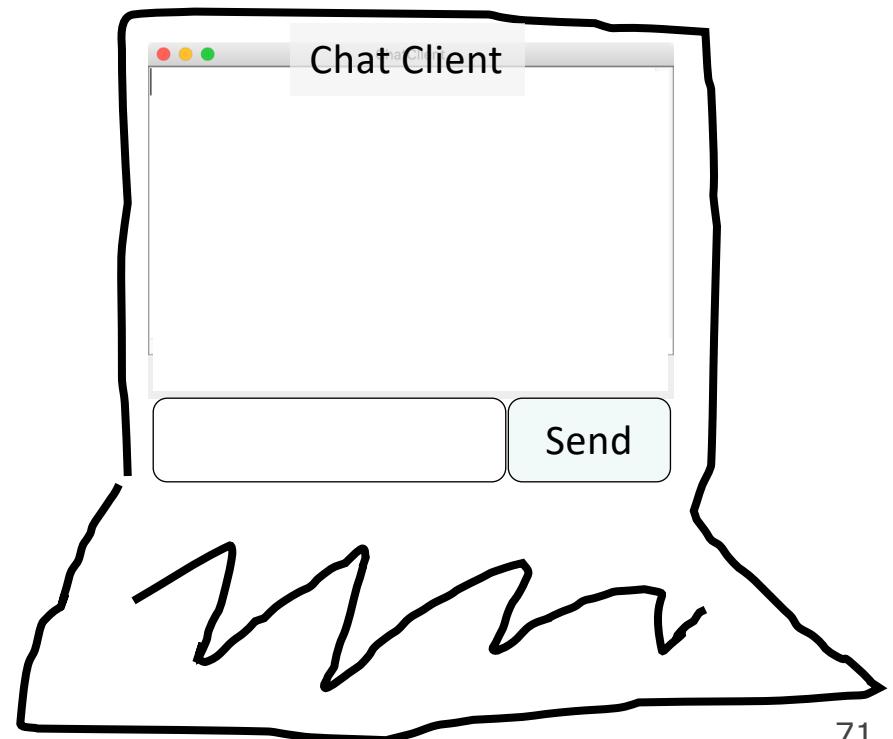
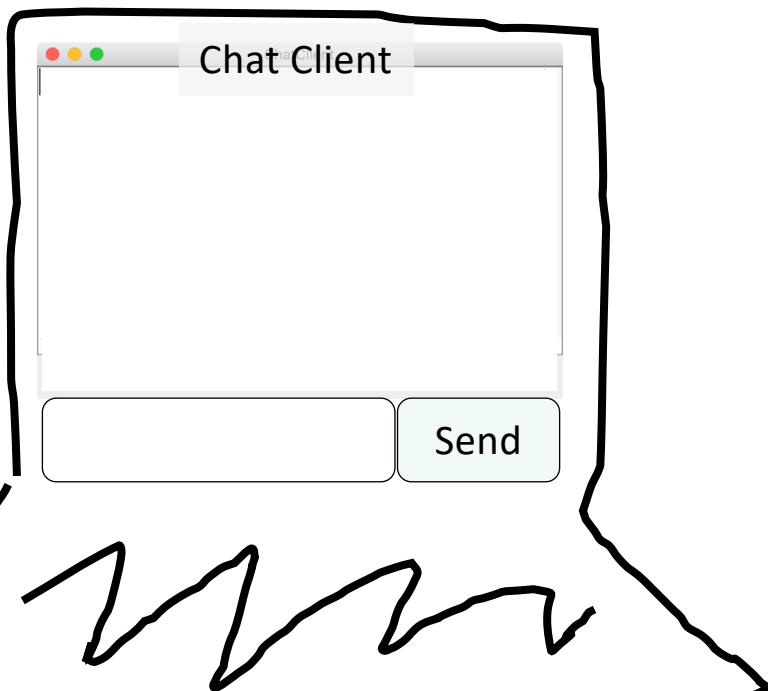
addMsg  
msg = C: Hello world

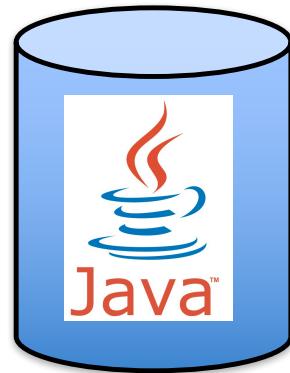




```
history = [  
    C: Hello world  
]
```

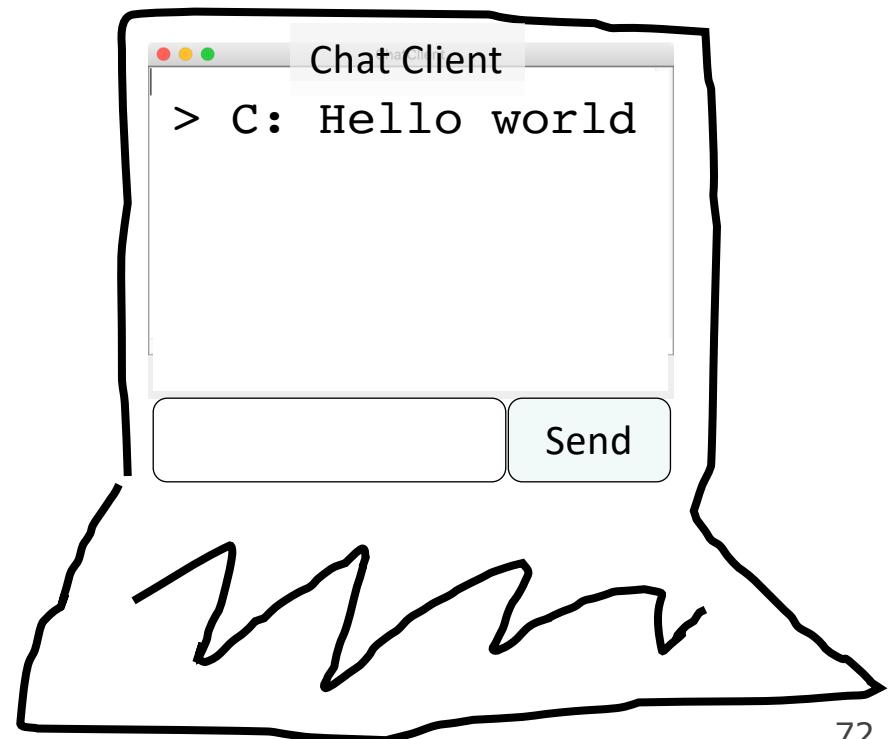
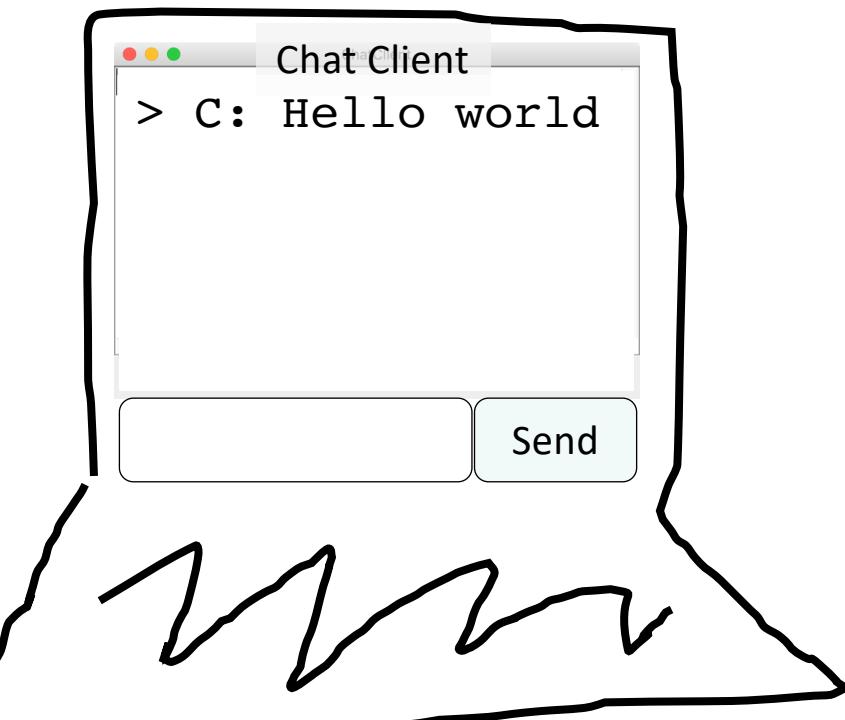
getMsgs  
index = 0

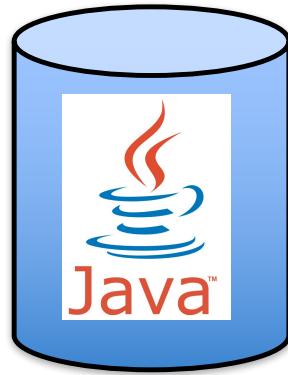




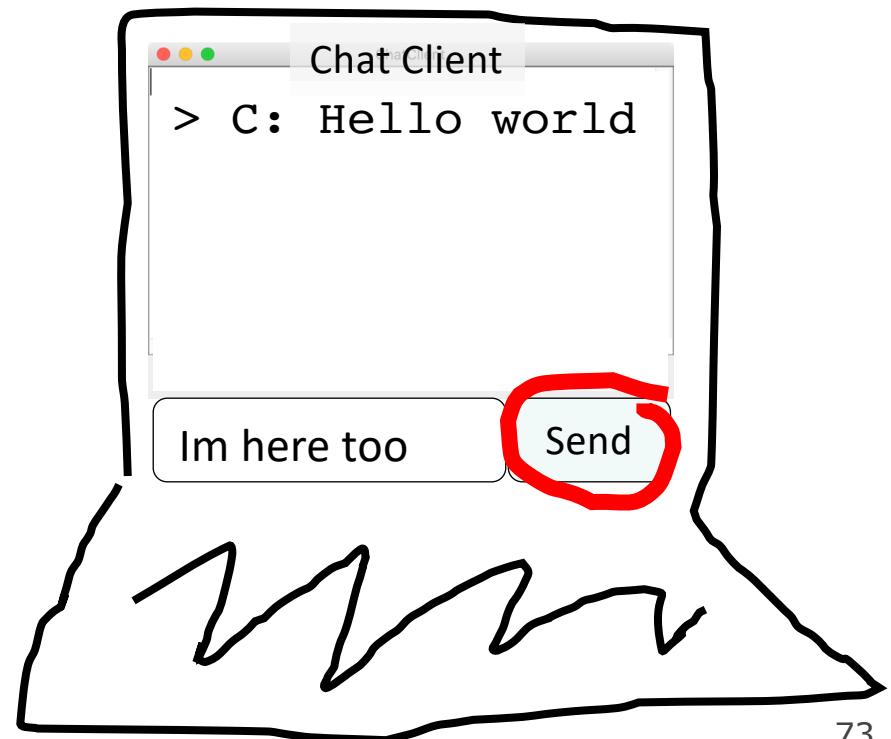
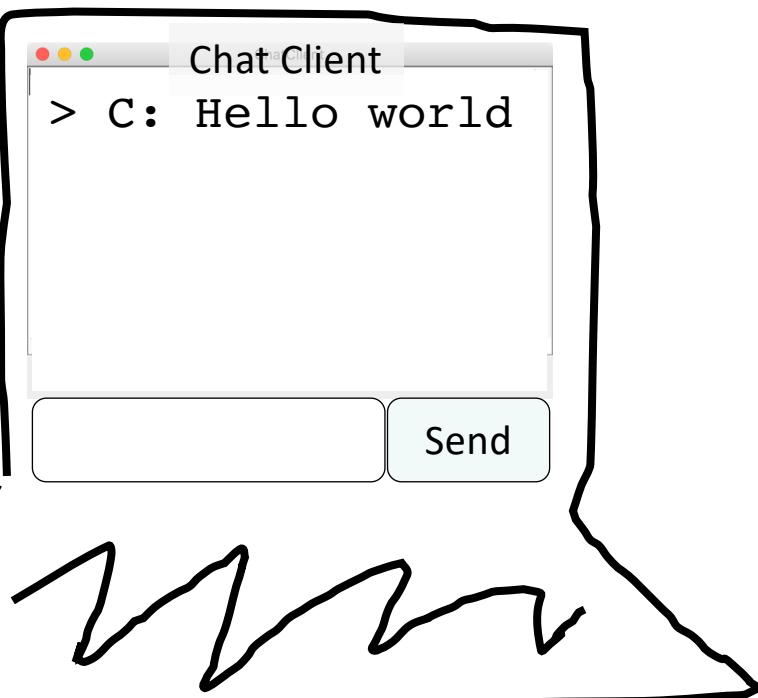
```
history = [  
    C: Hello world  
]
```

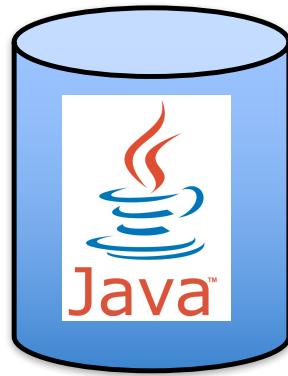
[C: Hello world]





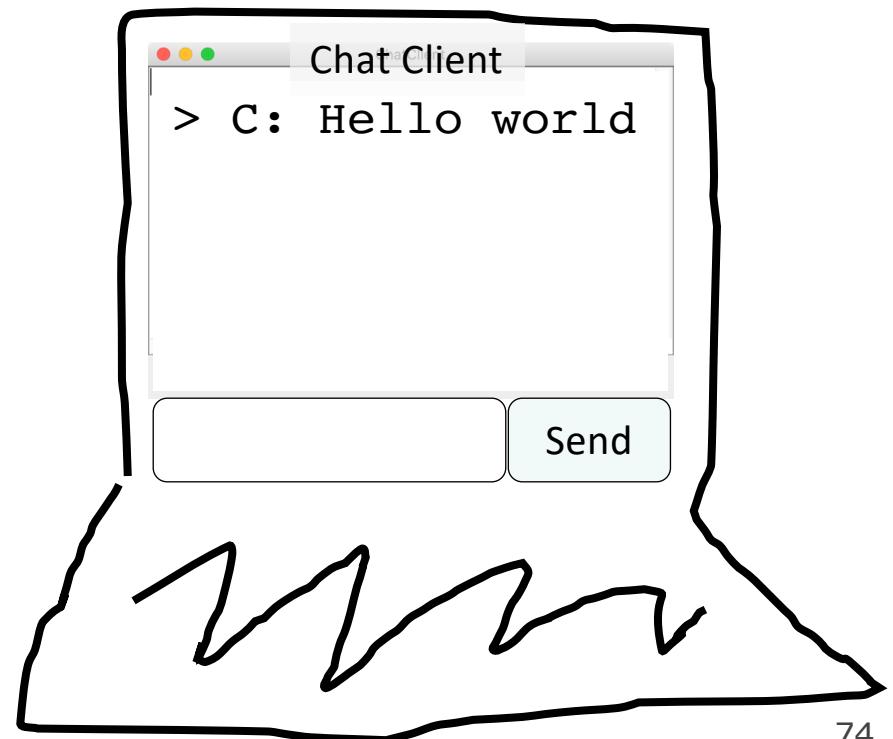
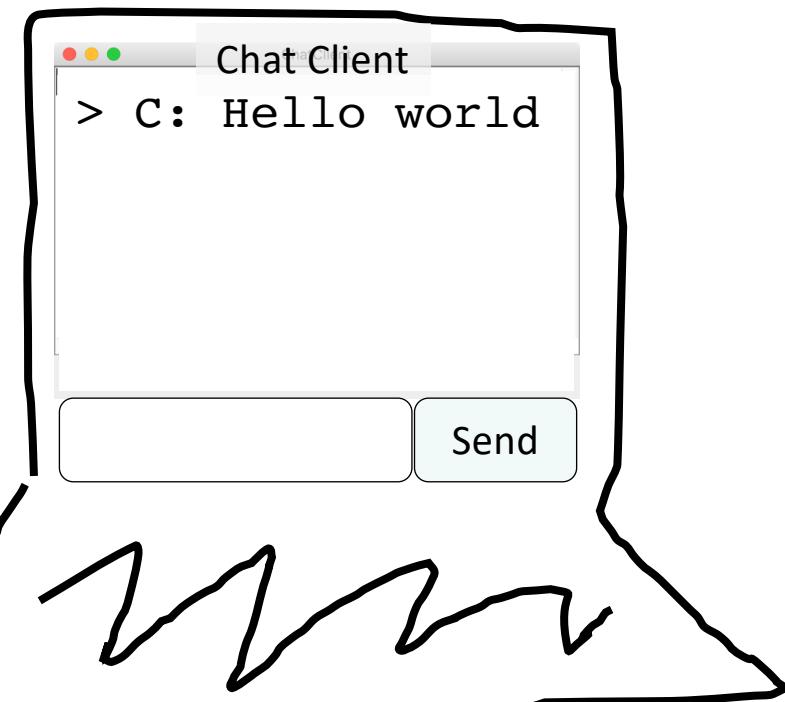
```
history = [  
    C: Hello world  
]  
  
addMsg  
msg = B: Im here too
```

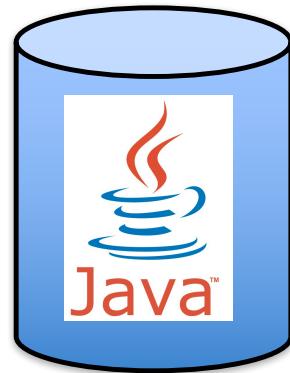




```
history = [  
    C: Hello world,  
    B: Im here too  
]
```

getMsgs  
index = 1

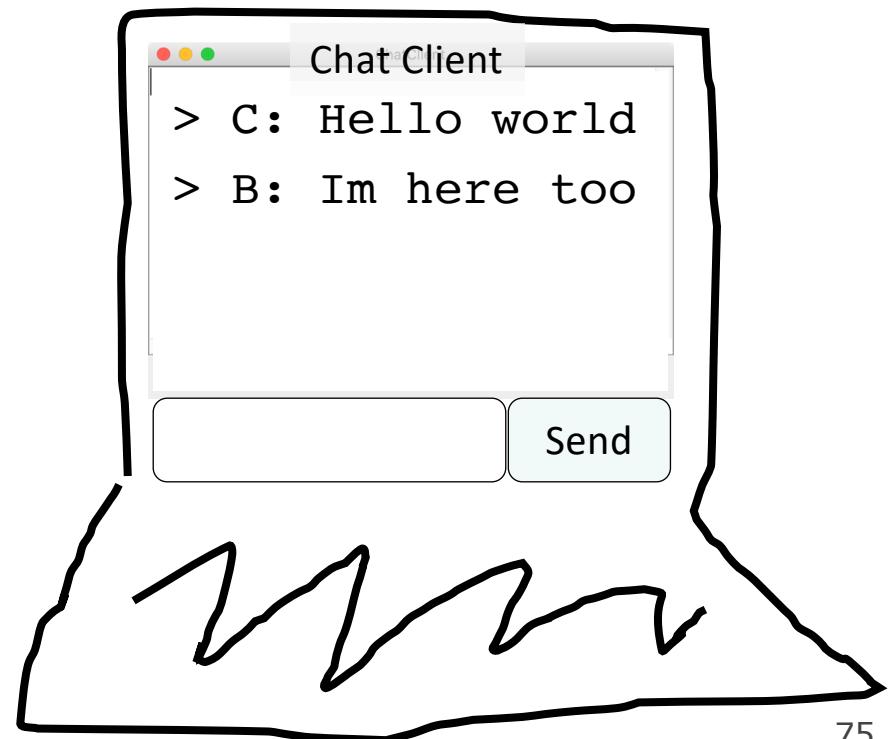
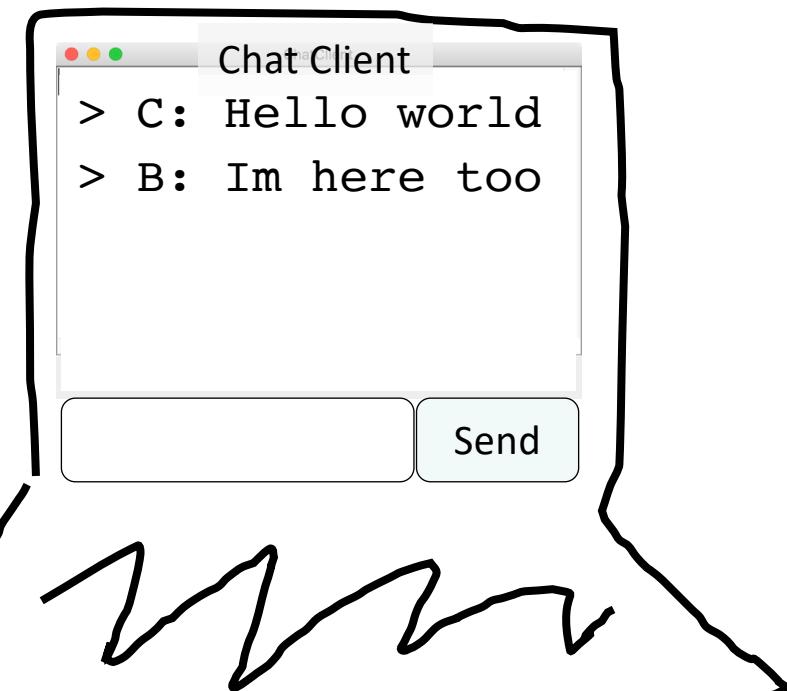




```
history = [  
    C: Hello world,  
    B: Im here too
```

]

[ B: Im here too ]

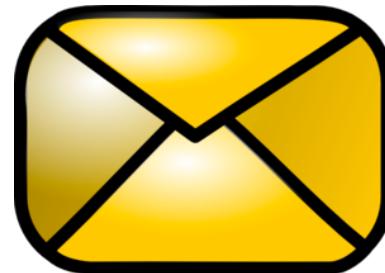


# Chat Server

Chat Server



addMsg  
msg = *text*



getMsgs  
index = *startIndex*



# **Let's Code It Together!**

# Recap

- How do internet programs work?
- Sending Data Over The Internet
- Writing a client and server
- *Demo:* echo
- *Demo:* Chat