Goals

- Introduce basic servlet terminology
- Formally introduce homework 2
- Walkthrough of build & deployment
- Group discussion of MMM

What is a Servlet?

- Server-side Java Program
- Equivalent to a CGI app
- Efficient
- Portable
- Powerful
- Stand-alone or extension
- We’re using Tomcat on cubist.cs.washington.edu

Demo Example

![Image](image.jpg)

GenericServlet

- Protocol Independent
- Override service() when extended
- Must write one’s own means of handling requests

Where do I find this stuff?

```java
/*For generic, protocol independent servlet functionality*/
import javax.servlet.*;
/*For servlets that will use the http protocol (WWW Based)*/
import javax.servlet.http.*/;

Tutorial @
https://java.sun.com/jakarta-jsp/docs/1.1/ServletAPIServlet.pdf

API @
```

Image from Java Servlet Programming by Jason Hunter
**HttpServlet**

- Extends \( \text{GenericServlet}() \)
- Override \( \text{doGet}(...) \) & \( \text{doPost}(...) \)
- Already understands http
- Majority of servlets extend

---

**HttpServlet Skeleton**

```java
import javax.servlet.*;
import javax.servlet.http.*;

public class ServletSkeleton extends HttpServlet {
    public void doGet(HttpServletRequest req,
                      HttpServletResponse res)
        throws ServletException, IOException {}
    public void doPost(HttpServletRequest req,
                        HttpServletResponse res)
        throws ServletException, IOException {
```

---

**GET vs POST**

<table>
<thead>
<tr>
<th>GET</th>
<th>POST</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieving information</td>
<td>send information</td>
</tr>
<tr>
<td>html/images</td>
<td>uploads/forms</td>
</tr>
<tr>
<td>parameters listed in URL</td>
<td>parameters hidden within header</td>
</tr>
</tbody>
</table>

---

**HW2 Prep Walkthrough**

Did you do your reading?

What project inspired Brooks to write this book?

Programming :) / :(?

What joys does computer programming have?

What woes?
Project Failure

What are some common reasons for project failure?

What is Brooks’?

All programmers are optimists:
"All will go well"

Scheduling

1/3 design
1/6 coding
1/4 component testing
1/4 system testing

Brooks’ Law

Adding manpower to a late software project will make it later.

Why?

The Surgeon Model

Fig. X.1: Communication patterns in 10-man programming teams