Outline

- Homework #3 solution
- Socket review
- HTTP Protocol
- URL and URLConnection
- Example
Sockets Review

- For applications to communicate, they need
  - a transport mechanism
  - an addressing mechanism
  - way to interact with these mechanisms

- For client/server applications
  - client = caller, originates requests
  - server = listener, handles requests
Sockets Review II

- Transport mechanism
  - TCP/IP
  - bi-directional data stream

- Addressing mechanism
  - hostname
  - port number
Sockets Review III

- Java classes
  - Socket for client
  - ServerSocket for server

- But
  - a call to accept returns with a regular Socket when a client connects
Socket Review IV

- Multi-threading
  - Extremely important for server applications
  - One thread listens continuously for connections
  - Other threads handle requests
HTTP Protocol

- Request
  - “I want something”

- Response
  - “Here it is”
  - or “Not found”

- Headers

- Body
HTTP Response Example

HTTP/1.1 200 OK
Date: Fri, 25 May 2001 22:51:05 GMT
Server: Apache/1.3.4 (Unix)
Set-Cookie: visit=1102002283923049566; path=/;
expires=Sat, 26 May 2001 22:51:05 GMT
Last-Modified: Mon, 21 May 2001 14:17:23 GMT
Accept-Ranges:bytes
Content-Length: 2000
Content-Type text/html

<HTML>
<HEAD>
<TITLE>All About Servers</TITLE>
</HEAD>
<BODY>
...

</BODY></HTML>
URL

- Example
  - http://www.cti.depaul.edu/people/

- Parts
  - Protocol
  - Host
  - Port
  - Path
In Java

- **URL classes**
  - new URL (http://www.cti.depaul.edu/

- **Simplest case**
  - url.getInputStream()
  - read from downloaded document
URLConnection

- `url.getConnection()`
- with URLConnection
  - initially “unconnected” state
  - can modify request properties/headers
- `connect()` or `getInputStream()`
  - “connected” state
  - can read response properties/headers
Example
Server Programming Example

- WeatherServer
- Protocol
  - one line commands: ADD and INFO
  - one line responses