Using TCP and IP to implement the web

User types: http://www.cse.wustl.edu/akg/cse131

Client

GET (url)

File comes back

Problem:
To connect to the server, need its IP address

CSE web server accepts TCP connections on port 80
When targeting a specific machine:

Source code ➔ compiler ➔ machine code ➔ load into memory ➔ executing process

When targeting a virtual machine:

JVM is another program

JVM

Source code ➔ compiler ➔ bytecode ➔ load into JVM ➔ interpreter bytecode ➔ executing process ➔ JVM
Native Code
---
- error localization?
- faster

Interpreted in VM
- portability — “write once — run anywhere”
- Security
  - verify that it’s legal code
  - can have signed code
  - can check operations at run time
    - disk access
    - network conn.

Adv.

Enables code sharing in a safe way
Applets are an example
Applets:  
- JVM inside a web browser  
- when a page contains class files, can run them in the browser.

HTML file

```html
<html>
<title>Ken's applet</title>
<h1>Here's my applet...</h1>
<img src="foo.gif"
<applet code="MyApplet.class" width=640 height=480>
<param name="Message" value="Welcome to CSE131!"
</applet>
</html>
```
import java.applet.Applet;

public class MyApplet extends Applet {
    String announcement;
    int x, y;
    public void init() {
        announcement = getParameter("message");
        x = getWidth() / 2;
        y = getHeight() / 2;
    }
    public void paint(Graphics g) {
        g.drawString(announcement, x, y);
    }
}
methods to override from Applet:

init() — called once upon load

class() — called upon load (after init) +
each time browser returns to the page

stop() — called when applet should "pause"
(when browser leaves the page)

public void paint(Graphics g)

public void destroy() — called when applet
is removed from the browser