Servlet Hello World

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

public class HelloWorld extends HttpServlet {
    public void doGet( HttpServletRequest request,
                       HttpServletResponse response )
        throws ServletException, IOException
    {
        PrintWriter out = response.getWriter();
        out.println("Hello World");
    }
}
```

Some Simple Observations

- Inherits from HttpServlet
- There's no `main()` method
- Who "runs" HelloWorld?
- `doGet()`
  - HttpServletRequest
  - HttpServletResponse → sent back to the client browser

Compilation – The Hard Way

- `javac` HelloWorld.java
  - Where is the servlet library?
  - How to include it in compilation?
  - Copy HelloWorld.class to the right directory

Compilation – The Slightly Smarter Way

- Set environment variable
  - .bashrc
- Use `-d` option of `javac`

Generate HTML

- Set content type to "text/html"
- Output a HTML page using `println`
  - `<html>`
  - `<head>`, `<title>`
  - `<body>`
- Validate
  - validator.w3c.org
  - Sometimes useful for debugging
Package

- The advantages of using packages
- The necessity of using packages
  - A Class without a package qualifier cannot be used in a class that belongs to a package (enforced in JDK 1.4)
  - All servlets generated from JSPs are in packages

Package Example

- Package and directory structure
  - cs320.stu31.HelloWorld
- package statement
- URL to access a packaged servlet

Servlet Life Cycle

- Loaded for the first time – `init()`
- Per request – `service()`
  - dispatch to doXxx()
- Unloaded – `destroy()`
  - When requested by the admin or manager user
  - When the servlet is idle for a long time

I/O Re-examined

- Java application
  - Input
    - Console, GUI
    - File
  - CLP, Resource
  - Output
    - Console, GUI
    - File
- Servlet
  - Input ??
  - Output ??

File I/O Example

- Read `groceries.txt`
- Display the content in a HTML table
  - `<table>`
  - `<tr>`
  - `<td>`

Servlet Debugging Tips

- Debugging servlets could be rather hard
- Section 3.8
  - Use print statements
  - Invest some time in a good IDE
  - JBuilder Developer
  - Netbeans
  - Follow good programming practice
  - Check out HTML source
  - Restart server