3.1 Basic Servlet Structure

Listing 3.1 outlines a basic servlet that handles `GET` requests. `GET` requests, for those unfamiliar with HTTP, are the usual type of browser requests for Web pages. A browser generates this request when the user enters a URL on the address line, follows a link from a Web page, or submits an HTML form that either does not specify a `METHOD` or specifies `METHOD="GET"`. Servlets can also easily handle `POST` requests, which are generated when someone submits an HTML form that specifies `METHOD="POST"`. For details on the use of HTML forms and the distinctions between `GET` and `POST`, see Chapter 19 (Creating and Processing HTML Forms).

Listing 3.1 ServletTemplate.java

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

public class ServletTemplate extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {

    // Use "request" to read incoming HTTP headers
    // (e.g., cookies) and query data from HTML forms.

    // Use "response" to specify the HTTP response status
    // code and headers (e.g., the content type, cookies).

    PrintWriter out = response.getWriter();
    // Use "out" to send content to browser.
  }
}
```

Servlets typically extend `HttpServlet` and override `doGet` or `doPost`, depending on whether the data is being sent by `GET` or by `POST`. If you want a servlet to take the same action for both `GET` and `POST` requests, simply have `doGet` call `doPost`, or vice versa.

Both `doGet` and `doPost` take two arguments: an `HttpServletRequest` and an `HttpServletResponse`. The `HttpServletRequest` lets you get at all of the *incoming* data; the class has methods by which you can find out about information such as form (query) data, HTTP request headers, and the client's hostname. The `HttpServletResponse` lets you specify *outgoing* information such as HTTP status codes (200, 404, etc.) and response headers (`Content-Type`, `Set-Cookie`, etc.). Most importantly, `HttpServletResponse` lets you obtain a `PrintWriter` that you use to send document content back to the client. For simple servlets, most of the effort is spent in `println` statements that generate the desired page. Form data, HTTP request headers, HTTP responses, and cookies are all discussed in the following chapters.

Since `doGet` and `doPost` throw two exceptions (`ServletException` and `IOException`), you are required to include them in the method declaration. Finally, you must import classes in `java.io` (for `PrintWriter`, etc.), `javax.servlet` (for `HttpServlet`, etc.), and `javax.servlet.http` (for `HttpServletRequest` and `HttpServletResponse`).

However, there is no need to memorize the method signature and import statements. Instead, simply download the preceding template from the source code archive at http://www.coreservlets.com/ and use it as a starting point for your servlets.