Chapter 1

An introduction to web programming with Java
Objectives

Knowledge

1. Name the software component that runs on the client of a typical web application.
2. Name the two software components that run on the server of a typical web application.
3. Distinguish between HTML and HTTP.
4. Distinguish between static web pages and dynamic web pages.
5. Name three approaches for developing Java web applications.
6. Describe the software components that are required for developing servlet and JSP applications.
7. List and describe the three layers of a typical Java web application.
8. In general terms, describe the use of these directories as defined by the J2EE specification: webapps, document root, WEB-INF, and WEB-INF\classes.
Objectives (continued)

9. Name two IDEs that can be used for developing Java web applications.

10. Name two web servers that can be used for developing Java web applications.

11. Name a popular database server that can be used for developing Java web applications.
The first page of a shopping cart application
The second page of a shopping cart application
Components of a web application

Client computer

Web browser

Internet connection

Server computer

Web server

Database server
How a web server processes static web pages

Client
Browser

Server
Web server
HTML file

HTTP request
HTTP response

How a web server processes dynamic web pages

Client
Browser

Server
Web server
Web application
Database server

HTTP request
HTTP response
Three approaches for developing Java web apps

Servlet/JSP

- Is a lower-level API that does less work for the programmer.
- Provides a high degree of control over the HTML/CSS/JavaScript that’s returned to the browser.

JSF

- Is a higher-level API that does more work for the programmer.
- Makes it more difficult to control the HTML/CSS/JavaScript that’s returned to the browser.

Spring Framework

- Is a higher-level API that does more work for the programmer.
- Provides a high degree of control over the HTML/CSS/JavaScript that’s returned to the browser.
Terminology

- The *Java Standard Edition (Java SE)* includes the *Java Development Kit (JDK)* and the *Java Runtime Environment (JRE)*.
- The *Java Enterprise Edition (Java EE)* specification describes how web servers can interact with all Java web technologies.
- *Servlets* store the Java code that does the server-side processing.
- *JavaServer Pages (JSPs)* store the HTML that defines the user interface.
- *JavaServer Faces (JSF)* provides a higher-level API that replaces both servlets and JSPs.
- *Java Persistence API (JPA)* is an API for working with databases.
The components of a servlet/JSP application

- **Client**
  - Browser

- **Server**
  - Web server and Servlet/JSP engine
    - Java Development Kit (JDK)
  - Database server

HTTP request flows from the client to the server, and the HTTP response flows back from the server to the client.
The architecture for a servlet/JSP application

Presentation layer
- HTML files
- JSP files

Business rules layer
- Servlets
- JavaBeans
- Other Java classes

Data access layer
- Data access classes

- Database
- Text files
- Binary files
- XML files
The directory structure for a web application

- tomcat
  - webapps
    - musicStore (the root directory for HTML and JSP files)
      - admin
      - cart
      - catalog
      - META-INF (content.xml file)
      - WEB-INF (the web.xml file)
    - classes (the root directory for Java classes)
      - music
        - business
        - controllers
        - data
        - util
    - lib (JAR files for Java class libraries)
# The directories and files for a web application

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(root)</td>
<td>Contains the HTML and JSP files.</td>
</tr>
<tr>
<td><code>WEB-INF</code></td>
<td>Contains the web.xml file and is not directly accessible from the web.</td>
</tr>
<tr>
<td><code>WEB-INF\classes</code></td>
<td>Contains the servlets and other Java classes for your application. Each subdirectory corresponds with the package for the Java class.</td>
</tr>
<tr>
<td><code>WEB-INF\lib</code></td>
<td>Contains any JAR files that contain Java class libraries that are used by the web application.</td>
</tr>
<tr>
<td><code>META-INF</code></td>
<td>Contains the context.xml file that configures the web application context.</td>
</tr>
</tbody>
</table>
The NetBeans IDE
Popular IDEs for Java web development

- NetBeans
- Eclipse
- IntelliJ IDEA
Two popular web servers

Tomcat

- Is a servlet/JSP engine that includes a web server.
- Is free, open-source, and runs on all modern operating systems.
- Is a popular web server for Java web applications.

GlassFish

- Is a complete Java EE application server.
- Is free, open-source, and runs on all modern operating systems.
- Provides more features than Tomcat.
- Requires more system resources than Tomcat.
Other popular web servers

- WildFly (formerly JBoss)
- Jetty
- Oracle WebLogic
- IBM WebSphere
A popular database server

MySQL

- Is a relational database server that implements most of the SQL standard.
- Is free, open-source, and runs on all modern operating systems.
- Is the most popular database server for Java web applications.

Other popular database servers

- PostgreSQL
- Oracle
An ISP that provides servlet/JSP web hosting
The FileZilla program