

Introduction to Internet Programming

Introduction

What is Application Program?

Program designed to perform specific function directly for the user

What is web application?

Application that is accessed via web browser over a network such as the internet or intranet

What is Internet Programming?

- Includes broad variety of technologies spanning diverse areas such as
 - Protocols for communication networks
 - Interfaces to databases
 - Programming of GUI

Overview of the Syllabus

- Unit 1
 - Java programming
- Unit 2
 - Website basics, HTML5, CSS3, Web 2.0
- Unit 3
 - Client and Server side programming
- Unit 4
 - PHP and XML
- Unit 5
 - Introduction to Ajax and Web services

Text & Reference Books

- TEXT BOOKS

- Deitel and Deitel and Nieto, “Internet and World Wide Web - How to Program”, Prentice Hall, 5 th Edition, 2011.
- Herbert Schildt, “Java-The Complete Reference”, Eighth Edition, Mc Graw Hill Professional, 2011.

- REFERENCES

- Stephen Wynkoop and John Burke “Running a Perfect Website”, QUE, 2nd Edition, 1999.
- Chris Bates, Web Programming – Building Intranet Applications, 3 rd Edition, Wiley Publications, 2009.
- Jeffrey C and Jackson, “Web Technologies A Computer Science Perspective”, Pearson Education, 2011.
- Gopalan N.P. and Akilandeswari J., “Web Technology”, Prentice Hall of India, 2011.
- Paul Dietel and Harvey Deitel, “Java How to Program”, , 8th Edition Prentice Hall of India.
- Mahesh P. Matha, “Core Java A Comprehensive Study”, Prentice Hall of India, 2011.
- 7. Uttam K.Roy, “Web Technologies”, Oxford University Press, 2011.

IP

- IP function: transfer data from **source** device to **destination** device
- IP source software creates a **packet** representing the data
 - **Header**: source and destination IP addresses, length of data, etc.
 - **Data** itself
- If destination is on another LAN, packet is sent to a **gateway** that connects to more than one network

Transmission Control Protocol (TCP)

- Limitations of IP:
 - No guarantee of packet delivery (packets can be dropped)
 - Communication is one-way (source to destination)
- TCP adds concept of a **connection** on top of IP
 - Provides guarantee that packets delivered
 - Provide two-way (**full duplex**) communication

TCP

- TCP also adds concept of a **port**
 - TCP header contains port number representing an application program on the destination computer
 - Some port numbers have [standard meanings](#)
 - Example: port 25 is normally used for email transmitted using the Simple Mail Transfer Protocol (SMTP)
 - Other port numbers are available first-come-first served to any application

Hypertext Transport Protocol (HTTP)

- HTTP is based on the request-response communication model:
 - Client sends a request
 - Server sends a response
- HTTP is a stateless protocol:
 - The protocol does not require the server to remember anything about the client between requests.

What a web client (web browser) will do?

- Normally implemented over a TCP connection (80 is standard port number for HTTP)
- Typical browser-server interaction:
 - User enters Web address in browser
 - Browser uses DNS to locate IP address
 - Browser opens TCP connection to server
 - Browser sends HTTP request over connection
 - Server sends HTTP response to browser over connection
 - Browser displays body of response in the **client area** of the browser window

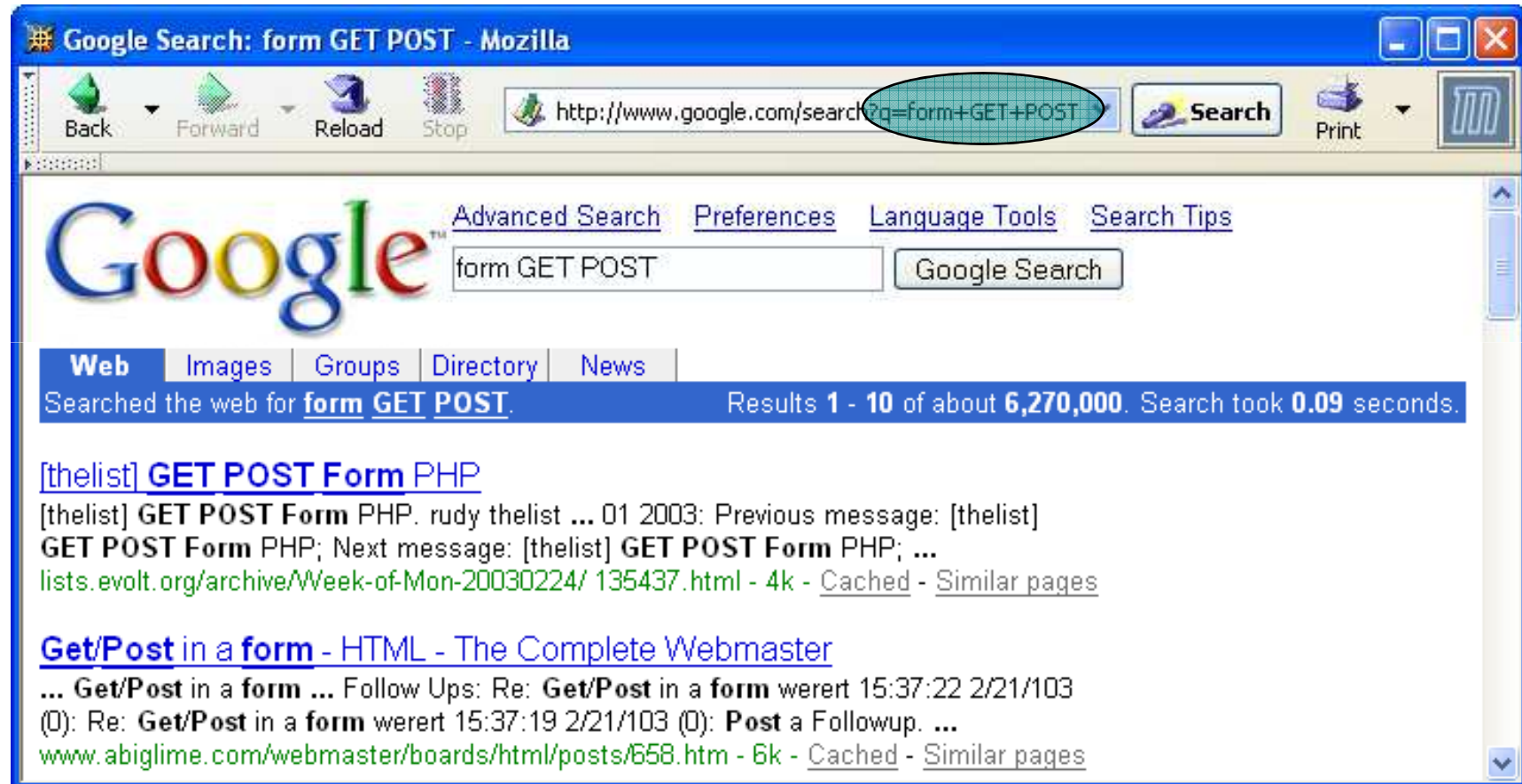
What a web server will do?

- Basic functionality:
 - Receive HTTP request via TCP
 - Map Host header to specific **virtual host** (one of many host names sharing an IP address)
 - Map Request-URI to specific resource associated with the virtual host
 - File: Return file in HTTP response
 - Program: Run program and return output in HTTP response
 - Map type of resource to appropriate MIME type and use to set Content-Type header in HTTP response
 - Log information about the request and response

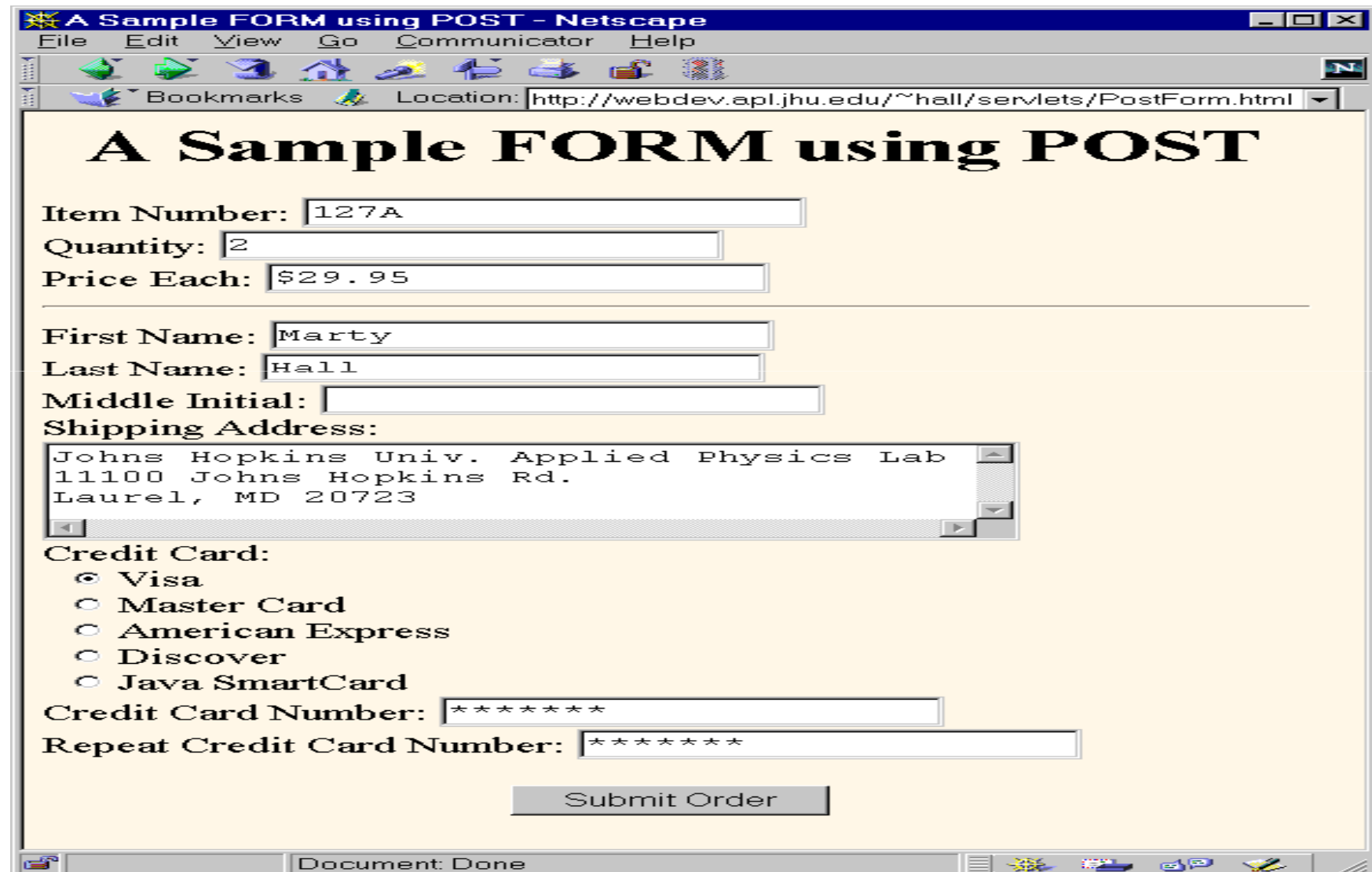
HTTP Request

- Common request methods:
 - GET
 - Used if link is clicked or address typed in browser
 - No body in request with GET method
 - POST
 - Used when submit button is clicked on a form
 - Form information contained in body of request
 - HEAD
 - Requests that only header fields (no body) be returned in the response

Sample Get Request



Sample Post Request



The screenshot shows a Netscape browser window titled "A Sample FORM using POST - Netscape". The address bar displays the URL "http://webdev.apl.jhu.edu/~hall/servlets/PostForm.html". The main content area has a yellow background and is titled "A Sample FORM using POST". It contains several form fields: "Item Number" (127A), "Quantity" (2), "Price Each" (\$29.95), "First Name" (Marty), "Last Name" (Hall), "Middle Initial" (empty), "Shipping Address" (Johns Hopkins Univ. Applied Physics Lab, 11100 Johns Hopkins Rd., Laurel, MD 20723), "Credit Card" (radio buttons for Visa, Master Card, American Express, Discover, Java SmartCard), "Credit Card Number" (*****), and "Repeat Credit Card Number" (*****). A "Submit Order" button is at the bottom. The status bar at the bottom says "Document: Done".

A Sample FORM using POST

Item Number:

Quantity:

Price Each:

First Name:

Last Name:

Middle Initial:

Shipping Address:
11100 Johns Hopkins Rd.
Laurel, MD 20723

Credit Card:

- ☒ Visa
- ☐ Master Card
- ☐ American Express
- ☐ Discover
- ☐ Java SmartCard

Credit Card Number:

Repeat Credit Card Number:

Document: Done

HTTP Request Header Fields

- Common header fields:
 - **Host**: host name from URL (required)
 - **User-Agent**: type of browser sending request
 - **Accept**: MIME types of acceptable documents
 - **Connection**: value `close` tells server to close connection after single request/response
 - **Content-Type**: MIME type of (POST) body, normally `application/x-www-form-urlencoded`
 - **Content-Length**: bytes in body
 - **Referer**: URL of document containing link that supplied URI for this HTTP request

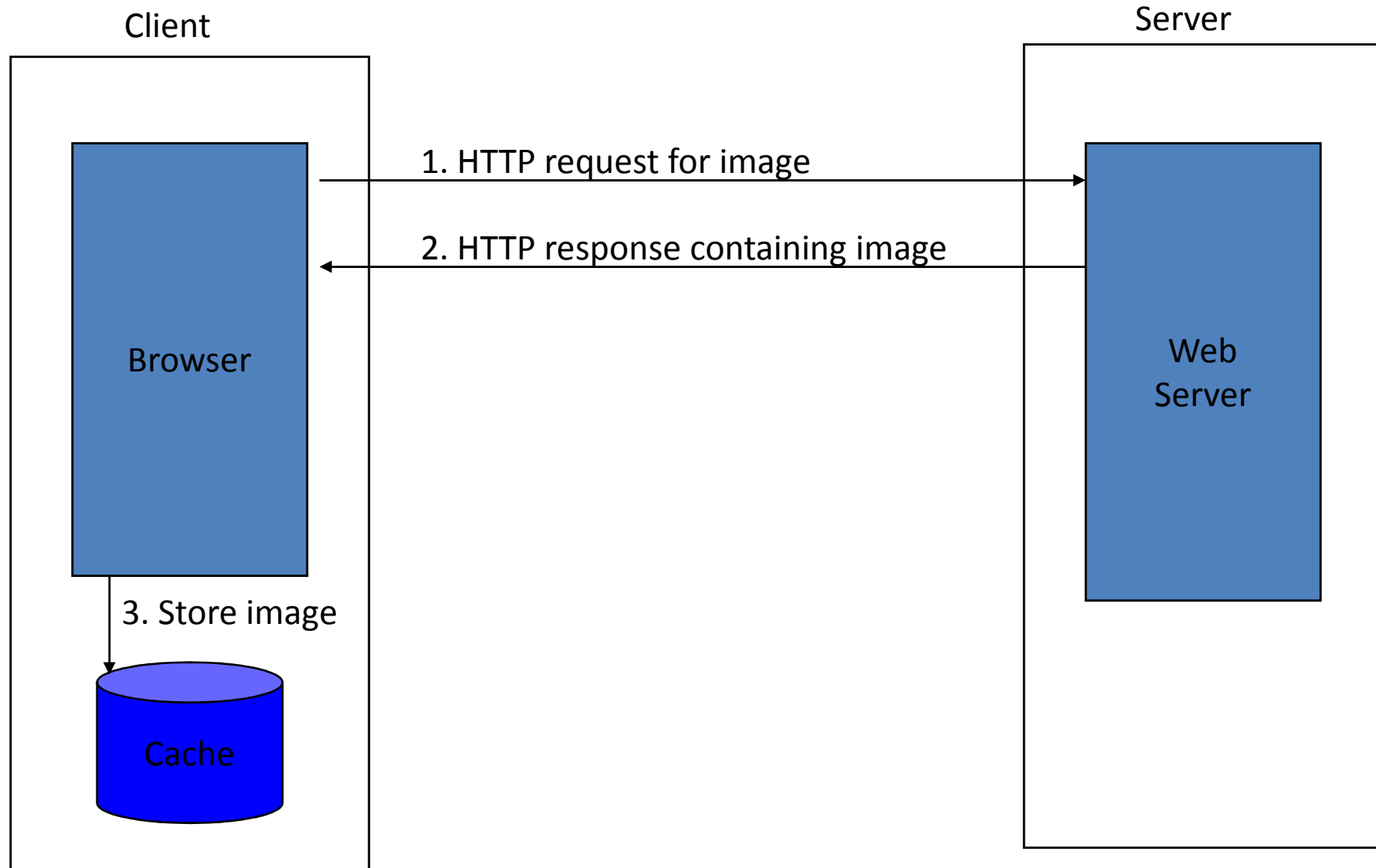
HTTP Response

- Status line
 - Example: HTTP/1.1 200 OK
- Three space-separated parts:
 - HTTP version
 - status code
 - reason phrase (intended for human use)
- Status code
 - Three-digit number
 - First digit is class of the status code:
 - 1=Informational
 - 2=Success
 - 3=Redirection (alternate URL is supplied)
 - 4=Client Error
 - 5=Server Error
 - Other two digits provide additional information

HTTP Response header fields

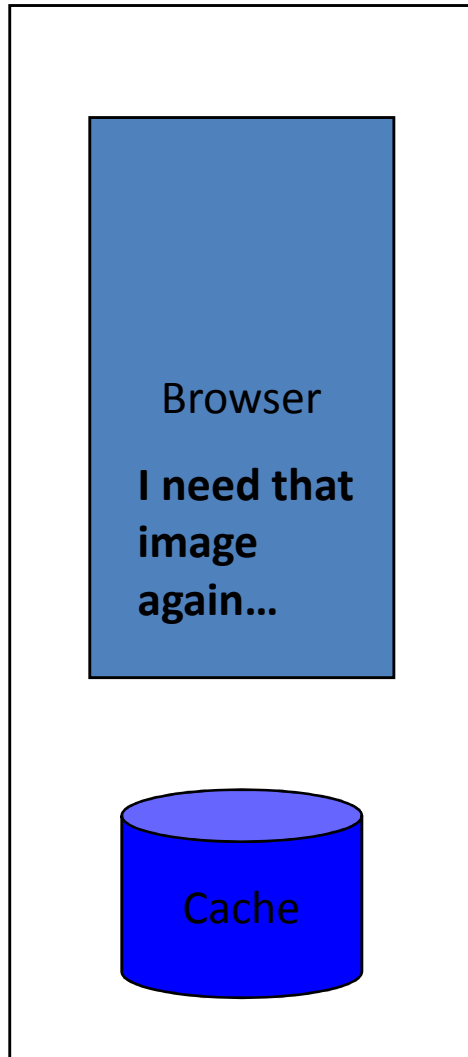
- Common header fields:
 - **Connection**, **Content-Type**, **Content-Length**
 - **Date**: date and time at which response was generated (required)
 - **Location**: alternate URI if status is redirection
 - **Last-Modified**: date and time the requested resource was last modified on the server
 - **Expires**: date and time after which the client's copy of the resource will be out-of-date
 - **ETag**: a unique identifier for this version of the requested resource (changes if resource changes)

Client Caching

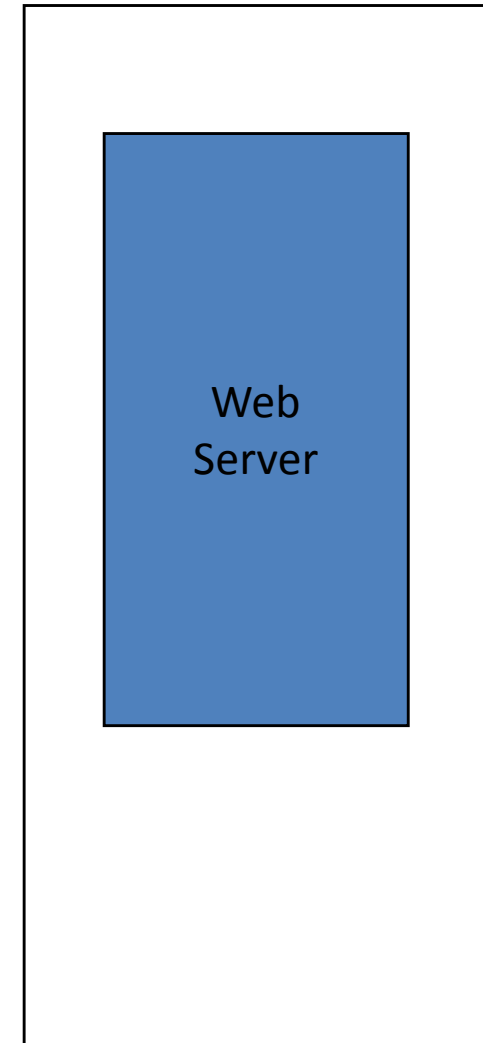


Client Caching

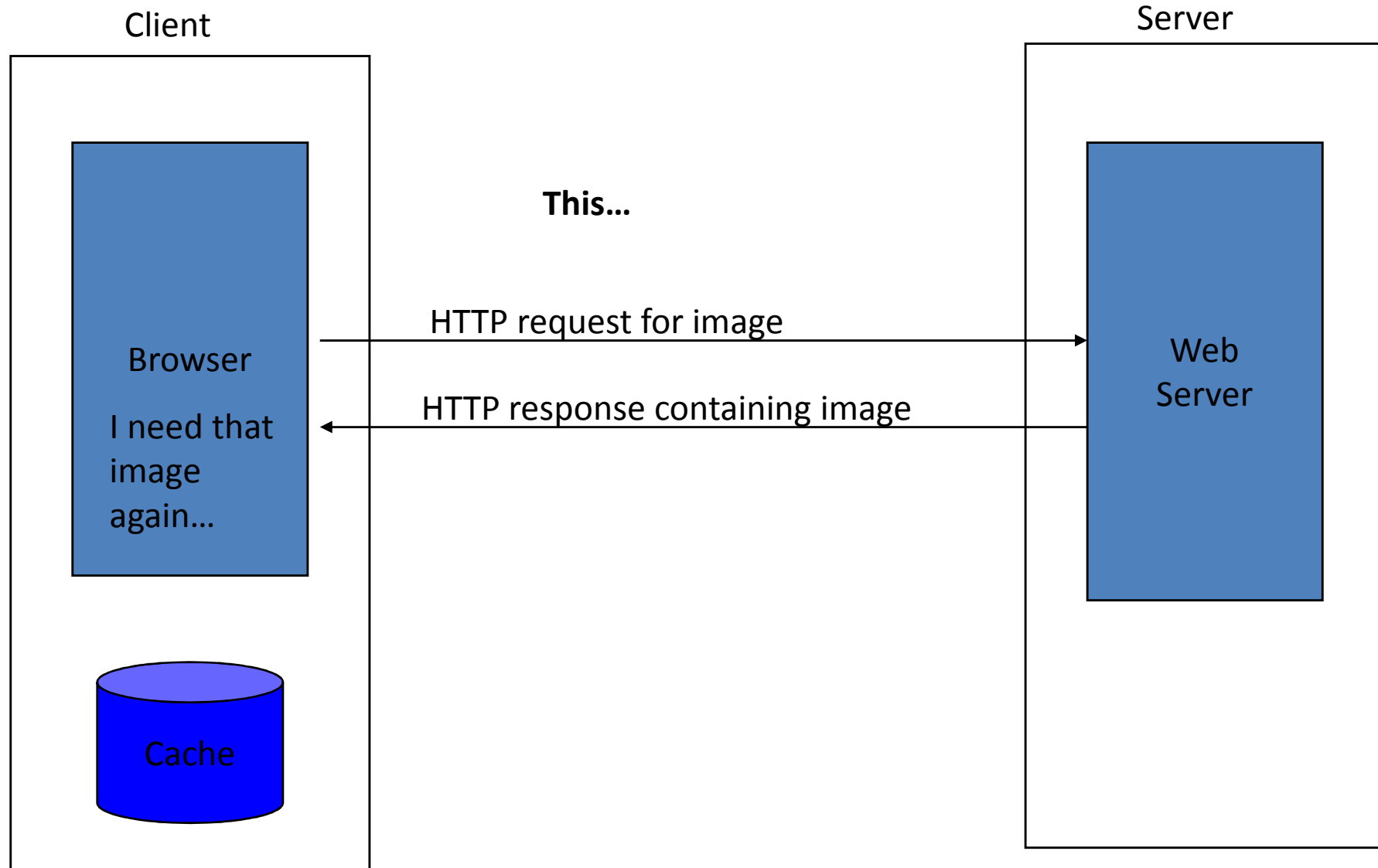
Client



Server



Client Caching



Course Outcomes

- Implement Java programs. **(K3)**
- Design a basic website using HTML, Cascading Style Sheets. **(K3)**
- Design and implement dynamic web page with validation using JavaScript objects and by applying different event handling mechanisms. **(K4)**
- Design rich client presentation using Servlet, AJAX and to present data in XML format. **(K4)**
- Design and implement server side programs using JSP, PHP. **(K4)**

