Web Software Model

CS 4640
Programming Languages for Web Applications

[Robert W. Sebesta, “Programming the World Wide Web”
Upsorn Praphamontripong, “Web Mutation Testing’’]
# Web Development (General View)

<table>
<thead>
<tr>
<th>Front End Development</th>
<th>Full-Stack Development</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$47k – $108k, avg = $71k</td>
</tr>
<tr>
<td>(developer / engineer)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$40k – $98k, avg = $71k</td>
</tr>
<tr>
<td>(developer / engineer)</td>
<td></td>
</tr>
</tbody>
</table>

- **Front End Development**
  - UI/UX design, usability
  - Web design
  - Responsive design
  - Client side
  - HTML, CSS, JS (and JS libraries, frameworks)
  - Fixed huge amount of data, no database interaction

- **Back End Development**
  - Server side
  - Speed, performance, scalability, security, availability, accessibility, reliability
  - Business logic
  - Java, PHP, Python, Ruby on Rails, and back end frameworks
  - Database interaction
  - Server administration
  - Database, data science

- **Full-Stack Development**
  - Variety of skills (both front end and back end)
  - Horizontal technology development (+)
  - Not expert in particular skill (-)

[Salaries reference: PayScale](https://www.payscale.com)  
Last accessed: 6-Jan-2020]
Web Applications

• User interactive software programs, \textit{deployed on a web server, accessed via a web browser}

• Use enabling technologies to
  • Make web site contents \textit{dynamic}
  • Allow users of the system to implement business logic on the server

• Let users \textit{affect state} on the server

• Constructed from diverse, distributed, and dynamically generated \textit{web components}
  • Web components are software modules that implement different parts of the application’s functionality

\textbf{An enabling technology makes web pages interactive and responsive to user input}
How do Web Apps fit in with the World Around Them?

Object type: Hotel

Object type: Car

Object type: Car

Object type: Car
Objects and Properties

Object type: Hotel

Properties
Name: Awesome  
Rating: 5  
Rooms: 70  
Bookings: 56  
Pool: true  
Gym: true

Object type: Car

Properties
Make: UVA1  
currentSpeed: 30mph  
Color: yellow  
Fuel: gasoline

Object type: Car

Properties
Make: UVA2  
currentSpeed: 20mph  
Color: red  
Fuel: gasoline

Object type: Car

Properties
Make: UVA2  
currentSpeed: 35mph  
Color: blue  
Fuel: gasoline
Objects and Events

**Object type: Hotel**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserve</td>
<td>reservation is made</td>
</tr>
<tr>
<td>Cancel</td>
<td>reservation is cancelled</td>
</tr>
</tbody>
</table>

**Object type: Car**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break:</td>
<td>driver slows down</td>
</tr>
<tr>
<td>Accelerate:</td>
<td>driver speeds up</td>
</tr>
</tbody>
</table>

**Object type: Car**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break:</td>
<td>driver slows down</td>
</tr>
<tr>
<td>Accelerate:</td>
<td>driver speeds up</td>
</tr>
</tbody>
</table>

**Object type: Car**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break:</td>
<td>driver slows down</td>
</tr>
<tr>
<td>Accelerate:</td>
<td>driver speeds up</td>
</tr>
</tbody>
</table>

**Object type: Car**

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break:</td>
<td>driver slows down</td>
</tr>
<tr>
<td>Accelerate:</td>
<td>driver speeds up</td>
</tr>
</tbody>
</table>

© Praphamontripong
Objects and Methods

Object type: Hotel

<table>
<thead>
<tr>
<th>Method</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>makeReservation()</td>
<td>increases value of bookings property</td>
</tr>
<tr>
<td>cancelReservation()</td>
<td>decreases value of bookings property</td>
</tr>
<tr>
<td>checkAvailability()</td>
<td>subtracts value of bookings property from value of rooms property and returns number of rooms available</td>
</tr>
</tbody>
</table>

Object type: Car

<table>
<thead>
<tr>
<th>Method</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>changeSpeed()</td>
<td>increases or decreases value of currentSpeed property</td>
</tr>
</tbody>
</table>
## All Together

### Object type: Hotel

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
<th>Method called</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserve</td>
<td>reservation is made</td>
<td>makeReservation()</td>
<td>Name: Awesome</td>
</tr>
<tr>
<td>Cancel</td>
<td>reservation is cancelled</td>
<td>cancelReservation()</td>
<td>Rating: 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>checkAvailability()</td>
<td>Rooms: 70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bookings: 56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pool: true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Gym: true</td>
</tr>
</tbody>
</table>

**Method**
- makeReservation(): increases value of `bookings` property
- cancelReservation(): decreases value of `bookings` property
- checkAvailability(): subtracts value of `bookings` property from value of `rooms` property and returns number of rooms available

### Properties
- Name: Awesome
- Rating: 5
- Rooms: 70
- Bookings: 56
- Pool: true
- Gym: true

### Object type: Car

<table>
<thead>
<tr>
<th>Event</th>
<th>Happens when</th>
<th>Method called</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break</td>
<td>driver slows down</td>
<td>changeSpeed()</td>
<td>Make: UVAI</td>
</tr>
<tr>
<td>Accelerate</td>
<td>driver speeds up</td>
<td>changeSpeed()</td>
<td>currentSpeed: 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Color: yellow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fuel: gasoline</td>
</tr>
</tbody>
</table>

**Method**
- changeSpeed(): increases or decreases value of `currentSpeed` property

**Properties**
- Make: UVAI
- currentSpeed: 30
- Color: yellow
- Fuel: gasoline
# Web Browsers and Objects

**Object type:** Web Browser

**Properties**

Location: [http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html](http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html)

---

## Spring 2020 — Syllabus

**Contact information**

Instructor: Uporn Praphamontripong  
Office: Rice Hall 206  
Office Hours: TBD (until they are settled, whenever my door is open)

**Teaching Assistants**

Simranjit Bhalia, Logan Hylton, Ryan Ritzo, Dennis Yang

TA office hours are held in Rice Hall or Ollson Hall most days; a schedule will be released soon. See the office hours tab for more on how to get their help.

We have fantastic teaching assistants for this course. Get to know them!

Our TAs are students too. Please only contact TAs via Piazza or in person during their schedule. Office hours, do not contact them via email.

**Class hours**

Tuesday / Thursday 2:00PM - 3:15PM, Ollson Hall 120

**Overview**

The way web software is built has been rapidly changing, presentation to information gathering to direct customer. The complexity of software has steadily been increasing. Many. Have you ever wondered what technologies, frameworks, and languages you should be familiar with? After you graduate, what languages will be popular or will be in great demand?

How will you keep up with web development technologies? According to the UVA CS Industrial Advisory Board meeting, companies prioritize the foundation of web development over any specific technologies or frameworks.

This course will help you see how fundamental concepts of web development can be applied to develop reliable and usable web software regardless of the technologies, languages, or frameworks. Although we put our emphasis on the concepts, you will demonstrate your competence in web development by developing dynamic web software with several commonly used programming languages and technologies.
**BOM: Browser Object Model**

**BOM** – collection of objects that the browser makes available to us for use with JavaScript

![Diagram of Browser Object Model](image)

- `window Object`
- `location Object`
- `history Object`
- `document Object`
- `navigator Object`
- `screen Object`

---

**DOM**

- `form Object`
- `link Object`
- `image Object`

---

Spring 2020 – University of Virginia  
© Praphamontripong
DOM: Document Object Model

Properties
- URL: http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html
- lastModified: 06/01/2019 10:19:23
- Title: CS4640:WebPL

Event
- Load
- Click
- Keypress

What it does
- write(): adds content to the document
- getElementById(): accesses an element of a given id attribute

Object type: Document

Contact information
Instructor: Uporn Praphamontripong
Office: Rice Hall 206
Office Hours: TBD (until they are settled, whenever my door is open)

Teaching Assistants
Simranjit Bhatia, Logan Hylton, Ryan Ritzo, Dennis Yang
TA office hours are held in Rice Hall or Olsson Hall most days of the week, for more on how to get their help.

We have fantastic teaching assistants for this course. Get to know them. Our TAs are students too. Please only contact TAs via Piazza or email.

Class hours
Tuesday / Thursday 2:00PM - 3:15PM, Olsson Hall 120

Overview
The way web software is built has been rapidly changing. As it uses presentation to information gathering to direct customer sales, the complexity of software has steadily been increasing.

Have you ever wondered what technologies, frameworks, or programming languages you should be familiar with? After you graduate, how will you keep up with web development technologies? Companies prioritize the foundation of web development over specific programming languages or technologies.

This course will help you see how fundamental concepts of usable web software regardless of the technologies, languages, or frameworks you choose. You will demonstrate your competence in web development and commonly used programming languages and technologies.

Do users look at web apps the way they are? Or do users look at web apps the way they think?
Contact information

Instructor: Upsorn Praphamontripong
Office: Rice Hall 206
Office Hours: TBD (until they are settled, whenever my door is open!)

Teaching Assistants
Simranjit Bhatia, Logan Hylton, Ryan Ritso, Dennis Yang

TA office hours are held in Rice Hall or Olsson Hall most days; a schedule will be released soon. See the office hours tab for more on how to get their help.

We have fantastic teaching assistants for this course. Get to know them!

Our TAs are students too. Please only contact TAs via Piazza or in person during their scheduled office hours, do not contact them via email.

Class hours
Tuesday / Thursday 2:00PM - 3:15PM, Olsson Hall 120

Overview
The way web software is built has been rapidly changing. As use of the world wide web has shifted from information presentation to information gathering to direct customer sales (e-commerce) to enterprise applications, the amount and complexity of software has steadily been increasing. Many new technologies and frameworks have emerged everyday. Have you ever wondered what technologies, frameworks, or architectural styles you should use? What programming languages you should be familiar with? After you graduate, what languages will be popular or will be in great demand? How will you keep up with web development technologies? According to the UVA CS Industrial Advisory Board meeting, companies prioritize the foundation of web development over any specific technologies or frameworks.

This course will help you see how fundamental concepts of web development can be applied to develop reliable and usable web software regardless of the technologies, languages, or frameworks. Although we put our emphasis on the concepts, you will demonstrate your competence in web development by developing dynamic web software with several commonly used programming languages and technologies.
Server Side Processing

UI implemented in a browser

HTTP Request

Web server

Container engine

Program components

Client

HTTP Response

XAMPP

HTML

data
General Design Issues

Separation of concerns: design goals of web app design

- Presentation
- Logic
- Data

Seven criteria the design should support:

- Reliability
- Usability
- Security
- Availability (and also Accessibility)
- Scalability
- Maintainability
- Performance and Time to market
Separation of Concerns

Presentation → HTML, view
Logic → Functions and classes, software
Data
- Data content
- Data representation
- Data storage → Classes, objects → Data structure, classes, objects → Database and files, Oracle, SQL, NoSQL

Assumptions about data
- Data values: contents of memory – change very frequently (during execution)
- Data presentation: how the data are presented – change occasionally
- Data structure: types, organization and relationships of different data elements, changes infrequently
Architectural Styles

• How to partition a system
• How components identify and communicate with each other
• How information is communicated
• How elements of a system can evolve

Web architectural styles constantly change

The goal is to separate logic from presentation and to separate as many concerns in the logic as possible
Multi-Tiered Architectures

Each software layer only communicates with adjacent layers

Presentation layer
- HTML, view

Business logic layer
- Classes, objects
- Web-specific libraries

Data layer
- Algorithms
- Business Processing

Data storage
- Data Reading and Writing
- DB, XML, Data files

Communicates with users

Most software is here

Separate concerns

Enables distribution
Requests are made to a main component and the main component response to clients – “client-server”
Requests are sent to a dispatcher that then forward the requests to another component (using forward or redirect control connection).
Model View Controller

• An abstraction frequently used in web app design

• Provide a way to divide the responsibilities of objects

• Decreases coupling between objects and layers (supports easier maintenance)

• Help divide the work (supports development expertise areas)
Model View Controller (2)

- **Model**
  - Encapsulates application state
  - Responds to state queries
  - Exposes application functionality
  - Notifies views of changes

- **View**
  - Renders the model
  - Requests updates from the model
  - Sends user inputs to controller
  - Allows controller to select view

- **Controller**
  - Defines application behavior
  - Maps user actions to model updates
  - Selects a view for response
  - One view for each function

[Graphic from Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition, Nicholas Kassem et al., October 2000]
Server-Side Scripting

- Generate HTML on the server through scripts
- Early approaches emphasized embedding server code inside HTML pages
- Examples: PHP, JSP

```php
<!doctype html>
<html>
<head>
<title>Login example</title>
</head>
<body>
You logged in as <font color="green">b><?php echo $_POST["name"]; ?>&gt;b></font><br>
with password <font color="green">b><?php echo $_POST["pwd"]; ?>&gt;b></font>
</body>
</html>
```

```jsp
<%@ page language="java" %>
<!-- Declare the variable -->
<%! int count = 0; %>
<!-- Scriptlet - Java code -->
<% for (int i = 0; i < 10; i++) {
    count = count+1;
}%>
<br />
The counter value is: <%= count %>
</jsp>`
Server-Side Scripting Site

Advantages
Server-side processing, browser independent, search optimization improvement, increased security

Disadvantages
Poor modularity, hard to understand, difficult to maintain
Server-Side Framework

- Structure server into tiers, organizes logic into classes
- Execution on the server
- Can be single-page or multiple pages with page-centric design, dispatcher design, model-view-control design, or combination of any architectures
- Example: JSP (with separation of concerns)

<table>
<thead>
<tr>
<th>Presentation</th>
<th>HTML and JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic</td>
<td>Java classes (non-servlet)</td>
</tr>
<tr>
<td>Data</td>
<td>Servlets and beans</td>
</tr>
<tr>
<td>Data content</td>
<td>Data structure, beans, Java classes</td>
</tr>
<tr>
<td>Data rep.</td>
<td>Database and files, Oracle, SQL</td>
</tr>
<tr>
<td>Data storage</td>
<td></td>
</tr>
</tbody>
</table>
Server-Side Framework Site

Advantages
Separation of concerns, maintain and reuse

Disadvantages
Need to load an entire page to get new data
Front-End Frameworks

- Client is organized into separate components, capturing model of web application data
- Components separate logic from presentation
- Components dynamically generate corresponding code based on component state
- Example: Angular
Advantages
Code organization, reuse, quick and easy to develop

Disadvantages
Duplicate logic in client and server
Single Page Application (SPA)

- Client-side logic sends messages to server, receives response

- Logic is associated with a single HTML page, written in JavaScript

- HTML elements dynamically added and removed through DOM manipulation

- Enabling technologies
  - AJAX
  - DOM Manipulation
  - JSON
  - Jquery
  - Angular (include and routing are useful)

- Example: Gmail
Single Page Application Site

Advantages

- Fast (load most resources once; only data are transmitted)

Disadvantages

- Hard to maintain and reuse, cross-site scripting, memory leak
Summary: Web Apps

• Deploy across the Web
  • Other (software) deployment methods include bundling, shrink-wrapping, embedding, and contracting

• New technologies

• New conceptual language constructs for programming
  • Integration
  • Data management
  • Control connections

• Browser features

These differences affect every aspect of how to engineer high quality software