华中师范大学伍伦贡联合研究院
Central China Normal University Wollongong Joint Institute

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

CSIT884
Web Development

Lecture 13 – Subject Review

Assessments

	Weight	Minimum for a Pass	Due Date	Submission
Assignments	40%	N/A	4 programming assignments with different due dates specified on Moodle	Via Moodle
Final Exam	60%	40% (24/60)	Exam Week **Dec 16, 2024**	
Total	100%	50%		

- Penalty will be enforced to late submissions according to Subject Outline.
- A mark of less than **40%** in Final Exam will result in a grade of **Technical Fail** for this subject.

Final Exam

- Exam arrangement
  - 3 Hours
  - Closed book
  - Question types
    - Mainly programming questions, with some short-answer questions
- Please be sure to review all topics covered in this subject, as any of them may appear in the exam

Final Exam

- Marking criteria
  - Correct programming syntax
  - Well-structured, clearly written and properly commented
  - Use proper data types and meaningful variable names
  - Non-redundant, efficient solution
  - **Correctness, completeness, and consistency with exam specifications**

Learning Objectives

1. Create web pages using a combination of HTML and JavaScript.
2. Use CSS to style a web page.
3. Understand and work with structured data formats JSON and XML.
4. Use XSLT to transform XML documents into other HTML format.
5. Design and create interactive web applications with AJAX.

HTML

- Learn the HTML (Hyper Text Markup Language) language to create web site
  - Document structure
- Understand the concept of HTML tags and attributes;
- Use HTML tags to format text, add graphics, create links, display tables, lists, etc
  - `<span>`
  - `<div>`
  - `<input type="..."  ...>, <textarea>,`
    - `select/option`
  - `<button>`

Web form

- Design a web form
  - text field, text area, checkboxes, radio buttons, option selection, submit buttons, ...
- Understand the web form HTML structure
  - name <-> value
- Use a web form to submit data to the server

CSS

- Understand the need of separation between the content and the style of your website
- Learn 3 different ways to define web page styles
- Use CSS to define styles for your web pages
  - `property:value`
  - `class, id`
  - CSS selectors

```
<h1 style="color:blue;">...</h1>
```

```
<style>
  h1 {color:blue;}
</style>
```

JavaScript

- Learn basic JavaScript programming language syntax
- Use JavaScript to make your website interactive
  - `<script>` tag
  - variables
  - functions
  - `if/if-else/for`
- String functions
- Date functions
- Arrays – create array and manipulate array elements

JavaScript events

- Understand JavaScript events
  - `onClick, onMouseOver, onChange, …`
- Write code to handle DOM events

```
<button onClick="handlerFunction()">
  Click me
</button>
```

- Confirm & Prompt boxes

JavaScript – Creating dynamic content

- Use JavaScript to build website with dynamic content
  - `document.getElementById("the-id")`
  - change the content of the HTML element
    - for span/div -> `el.innerHTML`
    - for input text field -> `el.value`
    - for image -> `el.src`
  - checkboxes & radio buttons -> checked vs value
- Use JavaScript to generate animation on your web site
  - `setInterval(animationFunction, milisecs)`
  - `clearInterval(animationSchedule)`
  - running only one animation at a time

CSS with JavaScript

- Use JavaScript to manipulate CSS property

```
var el = document.getElementById("the-id");
el.style.[cssProperty] = "the-new-style-value";
```

- Producing animated interactive effects with CSS properties
  - CSS Transitions - smooth out changes to property values between two states over time
  - CSS Transforms - allows to rotate, relocate, resize, and skew HTML elements in two- and three-dimensional space

Document Object Model - DOM

- Understand Document Object Model of a website
- Use JavaScript to generate dynamic content on a website
  - create/change/delete elements on the web

XML, DTD, XSD

- Use XML to store and transport data over the Internet
  - XML tags are defined by user
  - an XML document must contain one root element that is the parent of all other elements
  - XML elements needs to be properly nested
  - child element vs attribute
- Learn DTD language to define the structure of an XML document
- Learn XSD language to define the structure of an XML document

XSLT

- EXtensible Stylesheet Language Transformation (XSLT) is an XML language for transforming XML documents
  - used to transform XML file into other file formats, such as HTML
- describes how the XML elements should be displayed

```
<xsl:template match="…">
. . .
</xsl:template>
```

- get value

```
<xsl:value-of select="@attribute-name" />
<xsl:value-of select="element-name" />
```

- literal text

```
<xsl:text>. . . </xsl:text>
```

- for loop, if statement, choose-when-otherwise

JSON

- Understand the concepts of JavaScript array and object
- Use JSON to store and transport data over the Internet
- Translate JavaScript object to JSON format

```
objJSON = JSON.stringify(obj)
```

- Translate JSON data into JavaScript object

```
obj = JSON.parse(objJSON)
```

  - Use object properties to access the JSON data (display information on the web)

AJAX

- Understand the concept of asynchronous programming
- Write AJAX code to retrieve data from the server and display data on client web page

Graphic Canvas, Drag and Drop

- Use HTML5 Graphic Canvas feature for your webpage
  - `<canvas>` element, `getContext(…)` method
  - some methods: `fillText(…), strokeText(…), beginPath(…), closePath(…), moveTo(…), lineTo(…), stroke(…), fill(…), …`
  - some properties: `strokeStyle, fillStyle, lineWidth, …`
- Use HTML5 Drag and Drop feature to make your webpage interactive
  - Draggable elements, Droppable elements

Form Validation

- Built-in form validation
- Use JavaScript to perform form validation

Client-Side Storage

- Understand client-side web storage
- Write a program to store information into client-side web storage
- Write a program to read information from client-side web storage

Final Advices

- Revisit all lectures/examples & all lab exercises
  - Try to recreate and modify the examples
  - Check references/documentation for details when encountering problems
- Review all assignments
- If you have any questions or need clarification on specific topics, do not hesitate to reach out for support