# Web and Distributed Programming

Instructor: Eric Pogue
Session: 21


Agenda:

1. Friendly Conversation & Good Natured Banter… interaction coding vs lecture
2. Quick Review of Assignments From Last Class
3. Midterm Exam 1 Discussion
4. Google Maps Live Coding Session… TOGETHER!

Discussion & Questions welcome at any time… please be present with no phones or email during our discussion time

# Assignment From Last Class

Assignment:

- Work on Project 3 due Wednesday, Oct 23
- Compete Ch.6

# Discuss Midterm Exam #1

# Essay Question 1

**What are HTML, CSS, and JavaScript and what are the main uses for each? Please provide a detailed explanation.**

Answer:

HTML is the HyperText Markup Language. It is a language used for developing Web Pages. It works by marking up text of the document with tags that specify the structure and in some cases the presentation of the page to the browser.

CSS are Cascading Style Sheets. It is a language used for specifying the presentation of a document written using a markup language like HTML.

JavaScript is a scripting language used for developing dynamic web pages which can interact with the user. Common tasks that can be handled by JavaScript include validating user input, sending alerts to users, dynamically generating or changing HTML content, and game development.

# Essay Question 2

**What is a regular expression? What is its purpose and how can it be used within JavaScript?**

Answer: Regular expressions are a sequence of special characters which denote a pattern. They are used for matching various patterns in strings. Commonly, regular expressions are used for input validation and general text processing. In JavaScript, regular expressions are used for string methods that do pattern matching such as search, match, or replace. Alternatively, they can also be stored within a RegExp object and also used for pattern matching.

# Essay Question 3

**What is HTTP and how does it work? Provide a detailed explanation.**

Answer:

HTTP is the HyperText Transfer Protocol. It is an application layer protocol for the World Wide Web. It specifies how communication should be performed between a Web Browser and the Web Server. Using this protocol, the Web Browser sends a request message to the Web Server, indicating a request for a specific document. The Web Server receives this message, and sends back a response message with either the requested document or a dynamically generated document. The Web Browser then displays the contents of the response message to the user.

# Essay Question 4

**Write the HTML code that will insert a hyperlink image in a document. The image filename is “pic.jpg” and the hyperlink URL is “page.html”.**

Answer:

```
<a href = "page.html" >
   <img src="pic.jpg" />
</a>
```

# Other Questions

**Which of the following regular expression will find a match in the following string?**
**“Lewis University Fall 2012 CS 247-1”**

Answer: /\d\d\d/

Which of the following regular expression will find a match in the following string?
“Lewis University Fall 2012 CS 247-1”

Incorrect:
/247*-[a-z]/

Correct Answer:
/\d\d\d/

# Midterm Exam 1 Final Notes

- If you disagree on a question send me and email the provides the question, your answer, and why you feel the answer is correct
- Overall Grades do not look like they are calculating correctly in Blackboard… I am looking into why this is the case
- Per the Syllabus Midterm Exam 1 will be 10% of your final grade

# Todays Assignment

Assignment (before next class):

- Work on Project 3 due at the end of next week
- Complete Ch.7.1 through 7.4

# Live Coding Session*
# Google Maps API

# How to Completing your Projects this semester

1. Start early on project assignments
2. Establish a physical work environment that allows you to focus for extended periods of time
3. Become comfortable with your development environment/tools
4. Read or re-read the project assignment and related materials
5. Suggest changes to the assignment if you feel you have found a better way
6. Save and deploy working versions of your project regularly… this allows you to experiment without risking what you have already accomplished
7. Ask for help if you are stuck… often simply articulating the problem/question will lead you to finding your own answer
8. Look for similar examples… but write your own code that you understand
9. Come to class and participate in class exercises… ask questions during or after class
10. Review the textbook and review the lecture slides

** Invest the time!

# Coding Session Objectives

1. Become comfortable with your development environment/tools
   - Chrome Developer Tools
   - HTML and JavaScript Validator
2. Save and deploy working versions of your project regularly… this allows you to experiment without risking what you have already accomplished
   - Start with a working, tested, and validated Hello World
   - Make incremental enhancements in new files… or using source code control
   - Be prepared to submit what you have at any point
3. Look for similar examples… but write your own code that you understand
4. Only you can choose to invest the time… time spent earlier in the project is worth more than time spent at the end

# Start With Working, Tested, and Validated Hello World

Yahtzee Dice Roller to Hello World… Let's do this TOGETHER:

1. Copy 'yahtzee-dice-roller-extern-js.html' and 'yahtzee-dice-roller-extern-js.js' to working folder
2. Rename them to 'hello-world.html' and 'hello-world.js'
3. Update the JavaScript link in 'hello-world.html'
4. Test hello-world application… does it work?
5. Cut CSS code out of 'hello-world.html'… see the change in behavior?
6. Create a new file 'hello-world.css', move CSS code to external file, past CSS code into external file, and add link to 'hello-world.html'… does it work?
7. Add 'console.log()' message to 'function random()'… test it with Developer Tools! Does it work?
8. Success!

# Create A Safe Environment to Make Changes

Step 1 – Create initial Google Maps files:

1. Copy all three ‘hello-world’ files and name them ‘google-maps-step-01’ html, css, and js files
2. Update links in ‘google-maps-step-01.html’ to reflect new names
3. Update title to ‘Google Maps’
4. Update ‘console.log()’ message to reflect ‘Step 1’ and test new files
5. Test utilizing Google Developer Tools… does it work?
6. Success!

# Add Tutorial Code for a Map and Markers

Step 2 – Add a Google Map and two Markers:

1. Copy all three ‘step-01’ files to ‘step-02’ html, css, and js files
2. Update links in ‘google-maps-step-02.html’ to reflect new names
3. Update ‘console.log()’ message to reflect ‘Step 2’ and test new files… does it work?
4. Now add code from Google Maps tutorial to Step 2 files
5. Test it!... does it work?
6. Add a marker like they did in the second tutorial… does it work?
7. Add a second marker with a custom icon and Listener… does it work?
8. Success!

9. Time check…

# Add Idle Listener that Checks Zoomlevel and Bounds

Step 3 – Add Listener for Idle, get zoom level, check bounds, and remove Yahtzee code:

1. Copy all three ‘step-02’ files to ‘step-03’ html, css, and js files
2. Update links in ‘google-maps-step-03.html’ to reflect new names
3. Update ‘console.log()’ message to reflect ‘Step 3’ and test new files… does it work?
4. Add a listener for Idle that logs a console message with the current zoom level… does it work?
5. Success!

# Remove or Repurpose Yahtzee Code

Step 4 – Create step 4 files and know what you will need to do to submit project:

1. Copy all three ‘step-03’ files to ‘step-04’ html, css, and js files
2. Remove/repurpose Yahtzee code to create Instructions and Hints… does it work?
3. If you want, you can utilize ‘step-04’ to continue enhancing your solution
4. Do you know what it would take to finalize and submit ‘step-04’?

# End of Session

Course Number: CPSC-24700

Instructor: Eric Pogue