Lab 08 **CSC 470-03** **Spring 2019**

Objective

Practice using the `XMLHttpRequest` object with two styles of asynchronous programming: Continuation-Passing Style and Promises using `async`/`await`.

Description

Write and test an HTML file that loads and appends at least three HTML fragment files, asynchronously, in order, using JavaScript only.

Complete this task twice, first using a Continuation-Passing Style, and second using Promises invoked using async/await syntax. Both approaches should use an `XMLHttpRequest` object to load each HTML fragment file.

Both approaches should append HTML fragments to the body of the web page (see hints). Because this occurs twice, you should see each fragment twice on the page, in the proper order.

Hints

- You may use the `p1.html`, `p2.html`, `p3.html` fragment files and the `Lab08_start.html` file, provided.
- Compare the way we handled asynchronous loading of `Image`s with the asynchronous behavior of `XMLHttpRequest`. Both objects have `onload` and `onerror` event handlers. Pattern your solution after the `loadImage()`, `addAllCPS()`, `loadImagePromise()`, and `addAllAsyncAwait()` methods in `async_cats.html`.
- The text fragment from a successful XHR request may be accessed as `xhr.responseText`. Return this to the callback function after a successful request.
- Return `xhr.statusText` in the callback after a failed request.
- You may attach an HTML fragment returned from a successful XHR request to a web page using the command like: `document.body.innerHTML += fragment;`
- You may limit your XHR HTTP methods to `GET`.
- You may ignore errors when implementing your solution using the continuation passing style approach.
- You will need a simple HTTP server, like the one built-in to Python.

Requirements

- Create at least three small HTML fragment files to be loaded. Each file may be as simple as a single HTML tag, such as `<p>Paragraph 1</p>`.
- Create a new HTML page with a `<body>` tag that contains only a `<script>` tag for your code.
- Loading the page should load and append to the page all HTML fragments, twice, in order. The first time use your continuation-passing style solution and the second time use your async/await solution.
- Your entire script must be enclosed in an IIFE.
- You MUST enter header comments in your JavaScript code including (1) your name, (3) description and or purpose of the assignment.
- You MUST comment your code, explaining what you did in each section.
- Submit JavaScript and/or HTML files on Canvas under the appropriate assignment.