**Assignment 5 – Sequence Game**

**Setup**

Each week you will be asked to create a new folder under web-231 following a naming convention of “week-<number>.” If we are on week two, the folder name should be “week-2.” All files associated with the weekly assignment will be added to the appropriate weekly folder. All programs must be linked in the index.html landing page under the “Weekly Assignments” section. Projects will be linked under the “Projects” section of the index.html landing page. To be clear, **all** of the JavaScript, HTML, images, and CSS files associated with a weekly assignment must be placed under the appropriate weekly folder. The page title for all HTML files in this course must say “WEB 231 – Enterprise JavaScript I.” And, all HTML and CSS files must be valid HTML/CSS, tested through the WC3 validator. The links were provided during WEB 200 and were added to the index.html landing page. Also, the blue border around the provided images is to show they are images and should not be included in your submission. In other words, do not add a blue border around your work, unless the instructions explicitly ask for it.

**User interface styling and formatting requirements are located in the HTML, CSS, and JavaScript Requirements document.**

HTML: **<yourLastName>-sequence-game.html**

CSS: **<yourLastName>-sequence-game.css**

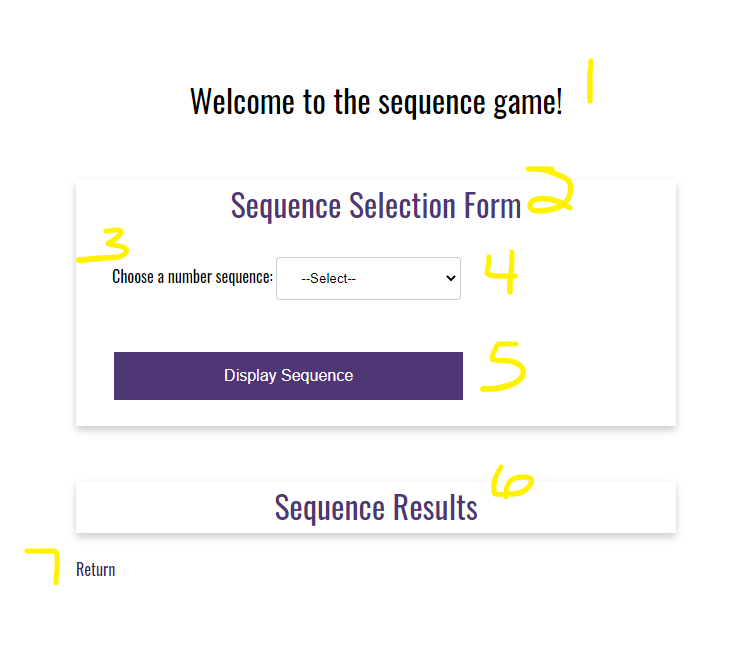
**Grading Reminders**

1. (50%-points) All code sources (.html, .css, .js) must be cited in the opening programmers’ comments, following the format specified in the code attribution document.
2. (25%-points) All code sources (.html, .css, .js) must show evidence of code comments. This means each section of the program (.html, .css, .js) must include code comments that explains what the block of codes purpose is, what the required parameters are (data type, if any), and what the expected output is.
3. (rubric) All code sources (.html, .css, .js) are measured against
   1. Code functionality: Does it work? Does it meet requirements?
   2. Adherence to standards and conventions. Are you using the appropriate data types, including proper indention, are variables named appropriate (variable x is an example of poor naming conventions), is there an appropriate use of whitespace, is the code organized, and are semicolons being used to terminate code sentences?
   3. Efficiency: Use of language features. Are you practicing DRY (Don’t-Repeat-Yourself?), are you leveraging built-in language features where appropriate, and are you using classes/functions to reduce code clutter?
   4. Documentation: Self-documenting, naming conventions, code is maintainable by others. Is the code your write easy to read and maintainable by others?
   5. Error trapping/handling. Are there errors in the program? Is there evidence of coding best practices to reduce user errors?
   6. Assignment Specific Compliance. Does the delivered solution follow the instructions, as they are written? Does the output match what was provided in the screenshots (including spaces, styling, etc.)?

**Required Modifications**

* Cite any sources in your opening programmer’s comment
* Link the appropriate CSS files and Google fonts

**Exhibit A. User Interface (Initial view)**

****

1. ~~h1 with a CSS class of app-header and a text value of “Welcome to the sequence game!”~~
2. ~~card-title with a text value of “Sequence Selection Form”~~
3. ~~form-field label with a text value of “Choose a number sequence”~~
4. ~~HTML select form-field with an id of~~ **~~sequence~~** ~~and a CSS class of “drop-down-menu”~~

**~~Additional Styling Requirements~~**

1. ~~Add a new class named~~ **~~drop-down-menu~~** ~~to the global.css file. Copy the values from the~~ **~~input~~** ~~class to this class.~~

**Additional HTML Requirements**

1. ~~The select form-field will need four HTML options~~
2. ~~value=select; text=--Select--~~
3. ~~value=fibonacci; text=Fibonacci Sequence~~
4. ~~value=even; text=Even Numbers~~
5. ~~value=odd; text=Odd Numbers~~

**Coding Assistance**

As the name suggestions, the value portion is assigned to the value attribute of the HTML option element. The text portion is the text that you will display in the select list. This week’s reading/resources section has an example of HTML select lists and option elements.

1. ~~HTML button with an id of btnDisplaySequence and a text value of “Display Sequence”~~

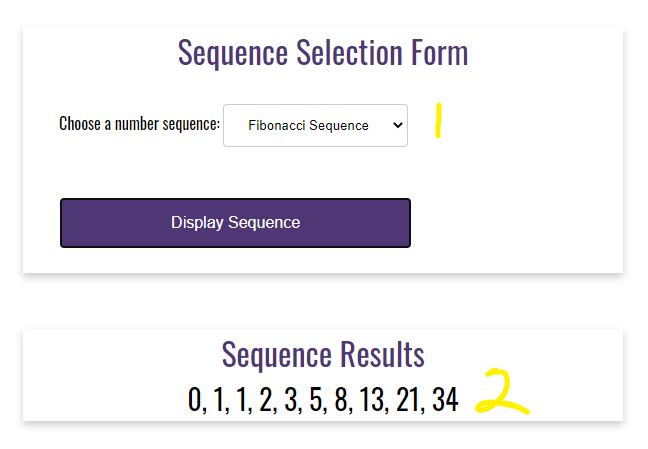
**Additional JavaScript Requirements**

1. Register an onclick event for btnDisplaySequence.
2. Capture the value from the select list and assign it to a variable named **choice**.
3. Using if…else if…else if…else, compare choice against the string values: fibonacci, even, and odd.
4. If fibonacci is selected (Exhibit B, #1), create a while loop that builds a string value of all fibonacci numbers that are less than 40. Bind the string to the results card-content div (Exhibit B, #2).

**Additional Programming Assistance**

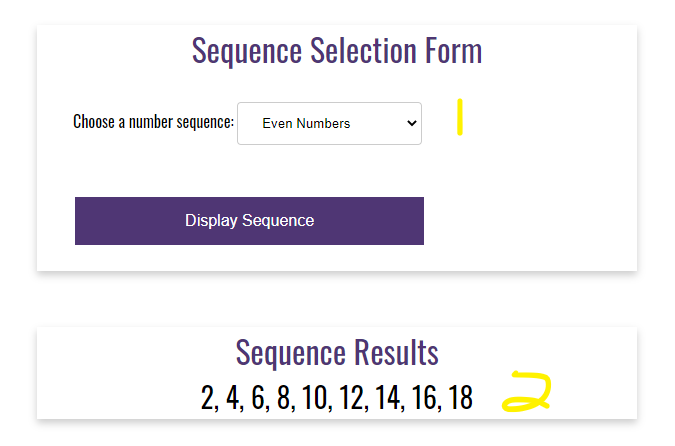
1. Define three variables to hold num1, num2, and next.
2. Define a variable to build the string output. Name it fibonacciSequenceText
3. while loop: update the variable fibonacciSequenceText with **num1 + “, “**
4. while loop: set next to num1 + num2
5. while loop: set num1 to num2
6. while loop: set num2 to next
7. Outside of the while loop, **bind** fibonacciSequenceText to the results card-content (Exhibit B, #2).
8. To remove the last two characters of the string use slice(0, -2). For example, fibonacciSequenceText.slice(0, -2);

* **Exhibit B. Fibonacci Sequence**



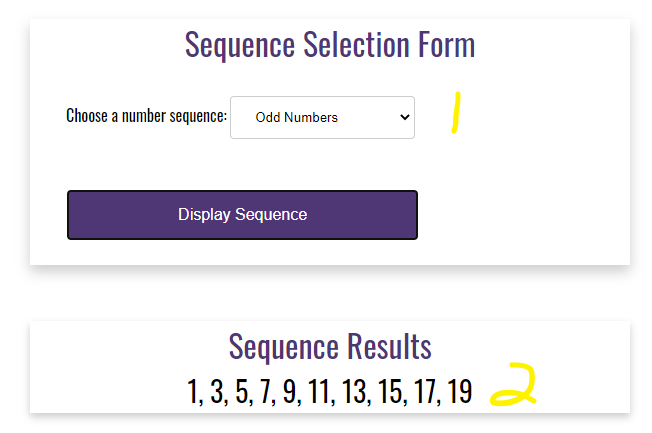
1. If even is selected (Exhibit C, #1), create a do…while loop that builds a string of even numbers that are less than 20. Bind the string to the results card-content div (Exhibit C, #2).

* **Exhibit C. Even Sequence**

****

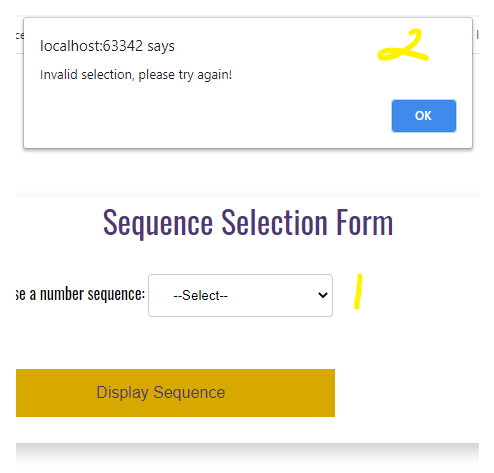
1. if odd is selected (Exhibit D, #1), create a for loop that builds a string of odd numbers that are less than 20. Bind the string to the card-content results div (Exhibit D, #2).

* **Exhibit D. Odd Sequence**



1. if “select” is chosen (Exhibit E, #1), alert “Invalid selection, please try again!” to the user (Exhibit E, #2)

* **Exhibit E. Invalid selection, please try again!”**



1. Styling

**Additional Styling Requirements**

1. As shown in Exhibits B, C, and D the text is centered with a font size of 28 pixels.
2. card-title with a text value of “Sequence Results”
3. anchor link with the CSS class return-home and a link back to the index.html landing page.