Assignment No. 4
EECS 468
Programming Language Paradigms
Due: 11:59 PM, Monday, October 2, 2023
Submit deliverables in a single zip file to Canvas
Files in other formats (e.g., .tar) will not be graded
Name of the zip file: FirstnameLastname_Assignment4 (with your first and last name)
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment4

Deliverables:

1. Copy of Rubric4.docx with your name and ID filled out (do not submit a PDF)
2. HTML source code.
3. The grader will execute your HTML file to ensure it runs.
4. Do not include any external files (e.g., .js, .css).
5. All of the JavaScript code must be included in the HTML file.

Assignment:

- Tabbed panels are widely used in user interfaces.
- They allow you to select an interface panel by choosing from a number of tabs “sticking out” above an element.
- In this assignment you must implement a simple tabbed interface using JavaScript and HTML.
- Write a function, asTabs, that takes a DOM node and creates a tabbed interface showing the child elements of that node.
- It should insert a list of <button> elements at the top of the node, one for each child element, containing text retrieved from the data-tabname attribute of the child.
- All but one of the original children should be hidden (given a display style of none).
- The currently visible node can be selected by clicking the buttons.
- Color the button for the currently selected tab red so that it is obvious which tab is selected.
- Color the rest of the buttons blue.
- The title of your web page (i.e., what shows up in the webpage tab title, not the tabs in your web page) should be “Your Last Name’s Go KU! Page”, where “Your Last Name” is your last name (e.g., “Johnson’s Go KU! Page”).
- The grader must be able to simply click on your source file and display the web page.
- On startup, the web page should display “Rock” highlighted and the text “Rock …” displayed
- Use this skeleton for your HTML file:

  ```
  <tab-panel>
      <div data-tabname="Rock">Rock ...</div>
      <div data-tabname="Chalk">Rock Chalk ...</div>
      <div data-tabname="Jayhawk">Rock Chalk Jayhawk ...</div>
      <div data-tabname="Go KU!">Rock Chalk Jayhawk - Go KU!</div>
  ```

```
</tab-panel>
<script>
  function asTabs(node) {
    // Your code here.
  }
  asTabs(document.querySelector("tab-panel"));
</script>
```

- Feel free to use the hints and sandbox from the Tabs exercise in Chapter 15. Note, this solution is a rather complicated program and if you use any part of it you will need to explain in the comments what it is doing in significant detail.
- Provide comments for the JavaScript code that explain what each line of code is doing. See rubric below. You do not have to comment the HTML code.

| **Rubric for Program Comments** | | |
|---|---|---|
| **Exceeds Expectations (90-100%)** | **Meets Expectations (80-89%)** | **Unsatisfactory (0-79%)** |
| Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line. | Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line. | Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line. |

Adequate Prologue Comments:

- Name of program contained in the file (e.g., EECS 468 Assignment 1)
- Brief description of the program, e.g.,
  - Hello World! examples using JavaScript and HTML
- Inputs (e.g., none, for a function, it would be the parameters passed to it)
- Output, e.g.,
  - Browser window with 2 test buttons
- All collaborators
- Other sources for the code ChatGPT, stackOverflow, etc.
- Author's full name
- Creation date: The date you first create the file, i.e., the date you write this comment

Adequate comments summarizing major blocks of code and comments on every line:

- Provide comments that explain what each line of code is doing.
- You may comment each line of code (e.g., using //) and/or provide a multi-line comment (e.g., using /* and */) that explains what a group of lines does.
- Multi-line comments should be detailed enough that it is clear what each line of code is doing.
- Each block of code must indicate whether you authored the code, you obtained it from one of the sources listed in the prolog, or one of your collaborators authored the code, or if it was a combination of all of these.

Collaboration and other sources for code:

- When you collaborate with other students or use other sources for the code (e.g., ChatGPT, stackOverflow):
    - Your comments must be significantly different from your collaborators.
    - More scrutiny will be applied to grading your comments in particular explaining the code “in your own words”, not the source’s comments (e.g., ChatGPT’s comments).
- Failure to identify collaborators or other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.
- Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.