

## Assignment No. 2

EECS 468

Programming Language Paradigms

Due: 11:59 PM, Monday, September 11, 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\_Assignment2 (with your first and last name)

Name of the Assignment folder within the zip file: FirstnameLastname\_Assignment2

### Deliverables:

1. Copy of Rubric2.docx with your name and ID filled out (do not submit a PDF)
2. HTML file named: Assignment2.html
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:

- Create in a single HTML file, the HTML and JavaScript code for the following modified exercises at the end of Chapter 2 in Eloquent JavaScript. (Note: If you use or modify the code in Chapter 2, which is perfectly fine, be sure to comment it as “code from other sources”).
- Your code should print out the name of the exercise before each exercise (e.g., “Looping a triangle”).
- Provide comments for both the HTML and JavaScript code that explain what each line of code is doing. See rubric below.
- Exercise 2.1 - Looping a triangle
  - Write a loop that makes 10 calls to output the following triangle:  
x  
xx  
xxx  
xxxx  
xxxxx  
xxxxxx  
xxxxxxx  
xxxxxxxx  
xxxxxxxxx  
xxxxxxxxxx
- Exercise 2.2 – FizzBuzz
  - Write a program that prints all the numbers from 1 to 100, with three exceptions:
    - For numbers divisible by 4, print "Divisible by 4" instead of the number.
    - For numbers divisible by 7, but not 4, print "Divisible by 7, but not 4" instead.
    - For numbers divisible by both 4 and 7, print “Divisible by both 4 and 7”.

- Exercise 2.3 – n-by-n Grid
  - Write a program that creates a string that represents an n-by-n grid, using newline characters to separate lines.
  - At each position of the grid there is either a space or a "\*" character.
  - The first position should be a space.
  - Define a binding name "size" so that it works for any size, outputting a square grid of the given width and height.
  - Show the output for size = 8 and size = 14.
  - Between the output for each size, print out the size (e.g. "size = 6").
  - For example, if size = 8, the grid would look like this:
 

```

* * * *
* * * *
* * * *
* * * *
* * * *
* * * *
* * * *
* * * *
          
```
- Hint: See the example of changing the text in a paragraph in the "JavaScript and the Browser" lecture.
- There are other more elegant ways to do this which we will learn about later in the semester. Feel free to explore these more elegant ways and use them. As long as your HTML/JS file produces the output described above you will get full credit.

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 368 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.