Assignment No. 3
EECS 468
Programming Language Paradigms
Due: 11:59 PM, Monday, September 18, 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_Assignment3 (with your first and last name)
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment3

Deliverables:

1. Copy of Rubric3.docx with your name and ID filled out (do not submit a PDF)
2. HTML file named: Assignment3.htm1
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:

- The standard JavaScript environment provides another data structure called Set.
- A set holds a collection of values (from Set Theory – EECS 210).
- A value can be part of a set only once—adding it again doesn't have any effect.
- Write a class called Group (since Set is already taken).
- Group should have 6 methods: add, delete, has, union, intersect, and difference.
- Its constructor creates an empty group.
- add adds a value to the group (but only if it isn't already a member)
- delete removes its argument from the group (if it was a member)
- has returns a Boolean value indicating whether its argument is a member of the group
- union returns the union of the group and the argument, which should be another group: this ∪ argument
- intersection returns the intersection of the group and the argument, which should be another group: this ∩ argument
- difference returns the difference of the group and the argument, which should be another group: this – argument
- To test your Group class, create an HTML file that does the following:
  1. Creates two groups as follows:
     ```
     let group1 = new Group();
     let group2 = new Group();
     group1.add(1);
     group1.add(2);
     group1.add(3);
     group2.add(2);
     group2.add(3);
     group2.add(5);
     group2.add(2);
     ```
  2. Then displays in separate paragraphs (using the <p> tag) for each of the following:

   a) Contents of group1
   b) Contents of group2
   c) Results of group1.has(5)
   d) Results of group2.has(3)
   e) Results of group1.union(group2)
   f) Results of group1.intersection(group2)
   g) Results of group1.difference(group2)
   h) Results of group1.delete(1)
   i) Results of group2.delete(1)

3. Each paragraph should include a label matching the description above, e.g., the label for a) should be "Contents of group1"

- Provide comments that explain what each line of JavaScript code is doing. You don't need to comment the HTML code. See rubric below.
- Do not use the built-in Set data structure in your code. Note: code for "union" that you find on the Internet may use the "Set" data structure.

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.