

Assignment No. 2

EECS 368

Programming Language Paradigms

Due: 11:59 PM, Monday, September 19, 2022

Submit deliverables in a single zip file to Canvas

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. JavaScript source code.
3. Screen print showing the successful execution of your JavaScript code. (Copy and paste the output from a web browser console screen to a Word document and PDF it).

Assignment:

- The standard JavaScript environment provides another data structure called Set.
- A set holds a collection of values (from Set Theory – EECS 210, see Sets & Set Operations.pptx for a review).
- 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: $\text{this} \cup \text{argument}$
- intersection returns the intersection of the group and the argument, which should be another group: $\text{this} \cap \text{argument}$
- difference returns the difference of the group and the argument, which should be another group: $\text{this} - \text{argument}$
- Test your Group class with the following code:

```
let group1 = new Group();
let group2 = new Group();
group1.add(1);
group1.add(2);
group1.add(3);
console.log(group1);
group2.add(2);
group2.add(3);
group2.add(5);
group2.add(2);
console.log(group2);
console.log(group1.has(5));
```

```

console.log(group2.has(3));
console.log(group1.union(group2));
console.log(group1.intersection(group2));
console.log(group1.difference(group2));
group1.delete(1);
console.log(group1);
group2.delete(1);
console.log(group2);

```

- Provide comments that explain what each line of code is doing. You may comment each line of code (using `//`) and/or provide a multi-line comment (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. 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.

Remember:

- Your Programming Assignments are individual-effort.
- You can brainstorm with other students and help them work through problems in their programs, but everyone should have their own unique assignment programs.