



Intro to JavaScript Week 5 Coding Assignment

Points possible: 70

| Category | Criteria | % of Grade |
|----------------------|---|------------|
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| Creativity | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| Completeness | All requirements of the assignment are complete. | 25 |

Instructions: In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

1. Create a menu app as seen in this week's video. What you create is up to you as long as it meets the following requirements.
 - a. Use at least one array.
 - b. Use at least two classes.
 - c. Your menu should have the options to create, view, and delete elements.



PROMINEO TECH

Screenshots of code:

```
<> menu.html U JS menu.js U X
JS menu.js > Menu > displayForms
1  class Student {
2    constructor(name) {
3      this.name = name;
4    }
5    describe() {
6      return `${this.name} is a student in form ${this.form}.`;
7    }
8  }
9
10 class Form {
11   constructor(formName) {
12     this.formName = formName;
13     this.students = [];
14   }
15   addStudent(student) {
16     if (student instanceof Student) {
17       this.students.push(student);
18     } else {
19       throw new Error(`You can only add an instance of Student. Argument is not a student: ${student}.`);
20     }
21   }
22   describe() {
23     return `Form ${this.formName} has ${this.students.length} students.`;
24   }
25 }
26
27 class Menu {
28   constructor() {
29     this.students = [];
30     this.selectedStudent = null;
31     this.forms = [];
32   }
```

```
<> menu.html U JS menu.js U X
JS menu.js > Menu > displayForms
33 start() {
34   let selection = this.showMainMenuOptions();
35   while (selection !== 0) {
36     switch (selection) {
37       case '1':
38         this.createForm();
39         break;
40       case '2':
41         this.viewForm();
42         break;
43       case '3':
44         this.deleteForm();
45         break;
46       case '4':
47         this.displayForms();
48         break;
49       default:
50         selection = 0;
51     }
52     selection = this.showMainMenuOptions();
53   }
54   alert('Goodbye!');
55 }
56
57 showMainMenuOptions() {
58   return prompt(`
59     0) Leave
60     1) Create a new Form
61     2) View Form
62     3) Delete Form
63     4) Display All Forms
64   `);
65 }
```



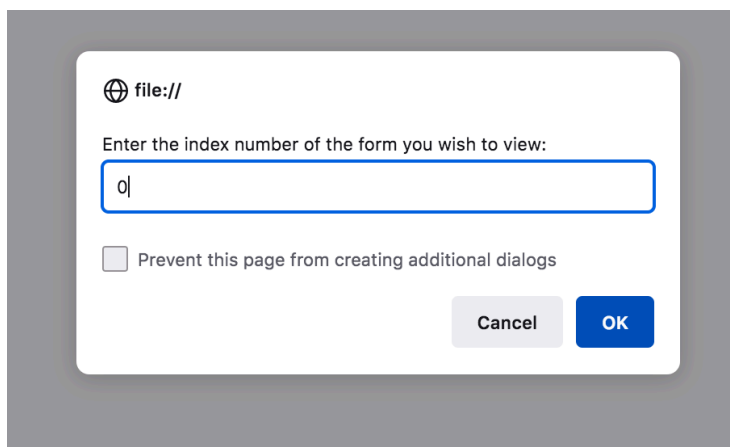
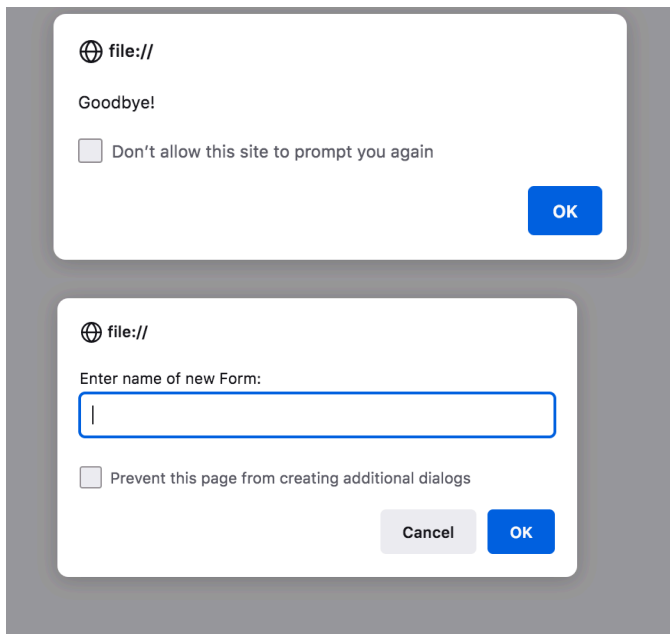
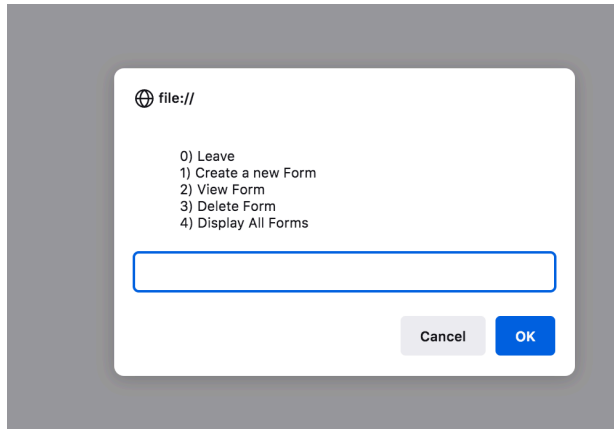
PROMINEO TECH

```
65     }
66     showFormMenuOptions(formInfo) {
67         return prompt(`
68             0) Back
69             1) Create Student
70             2) Delete Student
71             -----
72             ${formInfo}
73         `);
74     }
75
76     displayForms() {
77         let formString = '';
78         for (let i = 0; i < this.forms.length; i++) {
79             formString += i + ' ' + this.forms[i].formName + '\n';
80         }
81         alert(formString);
82     }
83
84     createForm() {
85         let name = prompt('Enter name of new Form:');
86         this.forms.push(new Form(name));
87     }
88
89     viewForm() {
90         let index = prompt('Enter the index number of the form you wish to view:');
91         if (index > -1 && index < this.forms.length) {
92             this.selectedForm = this.forms[index];
93             let description = 'Form Name: ' + this.selectedForm.formName + '\n';
94
95             for (let i = 0; i < this.selectedForm.students.length; i++) {
```

```
96                 description += i + ' ' + this.selectedForm.students[i].name + ' - ' + this.selectedForm.students[i].formName + '\n';
97             }
98
99             let selection = this.showFormMenuOptions(description);
100             switch(selection) {
101                 case '1':
102                     this.createStudent();
103                     break;
104                 case '2':
105                     this.deleteStudent();
106                     break;
107             }
108         }
109     }
110
111     deleteForm() {
112         let index = prompt('Enter the index number of the Form you wish to delete:');
113         if (index > -1 && index < this.forms.length) {
114             this.forms.splice(index, 1);
115         }
116     }
117
118     createStudent() {
119         let name = prompt('Enter name for new student:');
120         this.selectedForm.students.push(new Student(name));
121     }
122
123     deleteStudent() {
124         let index = prompt('Enter the number of the student you wish to delete:');
125         if (index > -1 && index < this.selectedForm.students.length) {
126             this.selectedForm.students.splice(index, 1);
127         }
128     }
129
130     et menu = new Menu();
131     enu.start();
132
133 }
```




Screenshots of Running Application:





PROMINEO TECH

 file://

0) Back

1) Create Student


2) Delete Student

Form Name: 2A

☐ Prevent this page from creating additional dialogs

Cancel

OK


 file://

Enter name for new student:

☐ Prevent this page from creating additional dialogs

Cancel

OK

 file://

0) 2A

☐ Don't allow this site to prompt you again

OK



PROMINEO TECH

Github URL: <https://github.com/zaccrmtree/week5codingassignment.git>