PROMINEO TECH

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.

URL to GitHub Repository: https://github.com/sw-dev-lisa-s-nh/JavaScript-Week5

```js
JS Week5MenuDrivenApp.js > ⚓ Menu > ⦿ start
 1   //
 2   //  JavaScript Week 5 Menu Driven App
 3   //   Lisa Maatta Smith
 4   //   due:  11/13/21
 5   //
 6
 7   class Instrument { // Like a Player
 8       constructor(name, section) {
 9           this.instrumentName = name;
10           this.section = section;
11       }
12
13       describe() {
14           console.log(`${this.instrumentName} is in the ${this.section} Section`)
15       }
16   }
17
18   class Musician { // Like a Team
19       constructor(name) {
20           this.name = name;
21           this.instruments = [];
22       }
23
24       addInstrument(instrument) {
25           if (instrument instanceof Instrument) {
26               this.instruments.push(instrument);
27           } else {
28               throw new Error(`You can only add an instance of Instrument.
29               This argument is not an instrument: ${instrument}`);
30           }
31       }
32
33       describe() {
34           console.log(`${this.name} plays ${this.instruments.length} instruments.`);
35           // POSSIBLY:  write a loop to get all of the instrument)
36       }
37   }
38
39   class Menu { // Menu class
40       constructor() {
41           this.musicians = []; // array of musicians
42           this.selectedMusician = null; // manage one musician at a time
43       }
44
45       start() { // entry point to application
46           let selection = this.showMainMenuOptions();
47           while (selection != 0) {
48               switch(selection) {
49                   case '1' :
50                       this.createMusician();
51                       break;
52
53                   case '2' :
54                       this.viewMusician();
55                       break;
56
57                   case '3' :
58                       this.deleteMusician();
59                       break;
60
61                   case '4' :
62                       this.displayMusicians();
63                       break;
64
65                   default:
66                       selection = 0;
67               }
68               selection = this.showMainMenuOptions();
69           }
70           alert('End of Musician Program!');
71       }
72
73       showMainMenuOptions() {
74           return prompt(`
75           Musician Main Menu:
76           ---------------------------------------
77               0) Exit
78               1) Create a new Musician
79               2) View a Musician
80               3) Delete a Musician
81               4) Display all Musicians
82           ---------------------------------------
83           `);
84       }
85
```

```
 85
 86     showMusicianMenuOptions(MusicianInfo) {
 87         return prompt(`
 88     Individual Musician Menu:
 89     -------------------------------------
 90         0) Return to Musician Menu
 91         1) Add a new Instrument
 92         2) Delete an Instrument
 93     -------------------------------------
 94         \t${MusicianInfo}
 95     `);
 96     }
 97
 98     displayMusicians() {
 99         let musicianString = '';
100         for (let i = 0; i < this.musicians.length; i++) {
101             musicianString += i+ ') ' + this.musicians[i].name + '\n';
102         }
103         alert(musicianString);
104     }
105
106     createMusician() {
107         let name = prompt('Enter the name for new Musician: ');
108         this.musicians.push(new Musician(name));
109     }
110
111     viewMusician() {
112         let index = prompt("Enter the index of the Musician that you want to view:");
113         if (index > -1  && index < this.musicians.length) {
114             this.selectedMusician = this.musicians[index];
115             let description = "\n Musician Name: " + this.selectedMusician.name + "\n";
116
117             for (let i = 0; i < this.selectedMusician.instruments.length; i++) {
118                 description +=  '\t\t' + i + ') ' + this.selectedMusician.instruments[i].instrumentName
119                     + ' - ' + this.selectedMusician.instruments[i].section + 'Section \n';
120             }
121
122             let selection1 = this.showMusicianMenuOptions(description);
123             switch (selection1) {
124                 case '1' :
125                     this.createInstrument();
126                     break;
127                 case '2' :
128                     this.deleteInstrument();
129             }
130
131         } else {
132             throw new Error(`Index: ${index} is an invalid Musician's index!`);
133         }  // validate user input
134     }
135
135     deleteMusician() {
136         let index = prompt('Enter the index of the Musician that you wish to delete: ');
137         if (index > -1  && index < this.musicians.length) {
138             this.musicians.splice(index,1);
139         } else {
140             throw new Error(`Index: ${index} is an invalid Musician's index!`);
141         }  // validate user input
142     }
143
144
145
146     createInstrument() {
147         let name = prompt('Enter name for new instrument: ');
148         let section = prompt('Enter section for new instrument: ');
149         this.selectedMusician.instruments.push(new Instrument(name, section));
150     }
151
152     deleteInstrument() {
153         let index = prompt('Enter the index of the instrument that you wish to delete: ');
154         if (index > -1 && index < this.selectedMusician.instruments.length) {
155             this.selectedMusician.instruments.splice(index,1);
156         } else {
157             throw new Error(`Index: ${index} is an invalid Instrument index!`);
158         }
159     }
160
161 }
162
163 let menu = new Menu();
164 menu.start();
165
```

Screenshots of Running Application:

Musician Main Menu:
----
    0) Exit
    1) Create a new Musician
    2) View a Musician
    3) Delete a Musician
    4) Display all Musicians

1

Cancel   OK

Enter the name for new Musician:

Minnie Mouse

Cancel   OK

Musician Main Menu:
----
    0) Exit
    1) Create a new Musician
    2) View a Musician
    3) Delete a Musician
    4) Display all Musicians

4

Cancel   OK

---

0) Minnie Mouse
1) Mickey Mouse
2) Daisy Duck
3) Donald Duck

Close

Musician Main Menu:
----
    0) Exit
    1) Create a new Musician
    2) View a Musician
    3) Delete a Musician
    4) Display all Musicians

2

Cancel   OK

Enter the index of the Musician that you want to view:

3

Cancel   OK

Individual Musician Menu:
----
    0) Return to Musician Menu
    1) Add a new Instrument
    2) Delete an Instrument

Musician Name: Donald Duck

1

Cancel   OK

Enter name for new instrument:

Tuba

Cancel   OK

Enter section for new instrument:

Low Brass

Cancel   OK

Individual Musician Menu:
----
    0) Return to Musician Menu
    1) Add a new Instrument
    2) Delete an Instrument

Musician Name: Donald Duck
    0) Tuba – Low Brass Section
      1) Trombone – Low Brass Section
      2) Piccolo – Woodwinds Section

2

Cancel   OK

Enter the index of the instrument that you wish to delete:

2

Cancel   OK

Individual Musician Menu:
----
    0) Return to Musician Menu
    1) Add a new Instrument
    2) Delete an Instrument

Musician Name: Donald Duck
    0) Tuba – Low Brass Section
      1) Trombone – Low Brass Section

Cancel   OK

Musician Main Menu:
----
    0) Exit
    1) Create a new Musician
    2) View a Musician
    3) Delete a Musician
    4) Display all Musicians

4

Cancel   OK

---

0) Minnie Mouse
1) Mickey Mouse
2) Daisy Duck
3) Donald Duck

Close

Musician Main Menu:
----
    0) Exit
    1) Create a new Musician
    2) View a Musician
    3) Delete a Musician
    4) Display all Musicians

3

Cancel   OK

Enter the index of the Musician that you wish to delete:

2

Cancel   OK

Musician Main Menu:
----
    0) Exit
    1) Create a new Musician
    2) View a Musician
    3) Delete a Musician
    4) Display all Musicians

4

Cancel   OK

---

0) Minnie Mouse
1) Mickey Mouse
2) Donald Duck

Close

End of Musician Program!

Close