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.0

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.

Screenshots of Code:

```js
class Ability {                          // A character's abilities and scores
    constructor(ability, score){
        this.ability = ability;
        this.score = score;
    }

    describe(){
        return `${this.ability} score is ${score}.`;
    }
}



class Characteristics {                  // A character's race and class
    constructor(race, rpgClass){
        this.race = race;
        this.rpgClass = rpgClass;
    }

    describe(){
        return `This character is a ${this.race} ${this.rpgClass}.`;
    }
}



class Character {                        // A character that can have a race, class, and abilities with scores
    constructor(name){
        this.name = name;
        this.characteristics = [];
        this.abilities = [];
    }

    addAbility(ability){                         // Allows an ability to be added
        if (ability instanceof Ability){
            this.abilities.push(ability);
        } else {
            throw new Error(`You can only add an instance of Ability. Argument is not an Ability: ${ability}`);
        }
    }

    addRaceAndClass(race, rpgClass){             // Allows race and class to be added
        if (race, rpgClass instanceof Characteristics){
            this.characteristics.push(race, rpgClass);
        } else {
            throw new Error(`You can only add an instance of Race and Class. Argument is not a Race or Class: ${race} ${rpgClass}`);
        }
    }
}
```

```javascript
49
50        describe(){
51            return `${this.name} is a ${race} ${rpgClass} and has the following abilities: ${this.abilities.join('\n')}`;
52        }
53    }
54
55
56    class Menu {
57        constructor(){
58            this.characters = []
59            this.selectedCharacter = null;
60        }
61
62        start(){
63            let selection = this.showMainMenuOptions();          // A main menu to start from where you can create, view,
64                                                                  //or delete characters, or display all characters made
65            while (selection != 0){
66                switch (selection){
67                    case '1':
68                        this.createCharacter();
69                        break;
70                    case '2':
71                        this.viewCharacter();
72                        break;
73                    case '3':
74                        this.deleteCharacter();
75                        break;
76                    case '4':
77                        this.displayCharacters();
78                        break;
79                    default:
80                        selection = 0;
81                }
82                selection = this.showMainMenuOptions();
83            }
84            alert("Good-bye!")
85        }
86
87        showMainMenuOptions(){                                   // Shows the main menu options
88            return prompt(`
89                0) Exit
90                1) Create New Character
91                2) View Character
92                3) Delete Character
93                4) Display All Characters
94            `);
95        }
96
97        showCharacterMenuOptions(characterInfo) {                // Shows the character menu options
```

```javascript
 96
 97    showCharacterMenuOptions(characterInfo) {              // Shows the character menu options
 98        return prompt(`
 99        0) Back
100        1) View Characteristics
101        2) View Abilities
102        ---------------------
103        ${characterInfo}
104        `);
105    }
106
107    showCharacteristicsMenuOptions(characterInfo){        // Shows the race and class menu options
108        return prompt(`
109        0) Back
110        1) Enter Race and Class
111        2) Change Race and Class
112        3) Delete Race and Class
113        ---------------------
114        // ${characterInfo}
115        `);
116    }
117
118    showAbilityMenuOptions(characterInfo){                // Shows the ability menu options
119        return prompt(`
120        0) Back
121        1) Create Ability
122        2) Change Ability
123        3) Delete Ability
124        ---------------------
125        ${characterInfo}
126        `);
127    }
128
129    joinCharacteristics(){                                // Joins the characteristics of a character into a string so that
130    var description1 = '';                                // you can see it in later code without having to reuse code
131        for (let i = 0; i < this.selectedCharacter.characteristics.length; i++){
132            description1 += i + ') ' + this.selectedCharacter.characteristics[i].race + ' '
133            + this.selectedCharacter.characteristics[i].rpgClass + '\n';
134        }
135        return description1;
136    }
137
138    joinAbilities(){                                      // Joins the abilities of a character into a string so that
139        var description2 = '';                            // you can see it in later code without having to reuse code
140        for (let i = 0; i < this.selectedCharacter.abilities.length; i++){
141            description2 += i + ') ' + this.selectedCharacter.abilities[i].ability
142            + ' - ' + this.selectedCharacter.abilities[i].score + '\n';
143        }
144        return description2;
```

```
144            return description2;
145        }
146
147
148        displayCharacters(){                                          // Joins all the characters made by the user in one string in
149            let characterString = '';                                 // order to display them together
150            for (let i = 0; i < this.characters.length; i++){
151                characterString += i + ') ' + this.characters[i].name + '\n';
152            }
153            alert(characterString);
154        }
155
156        isItUndefined(){                                              // Tests to see if either of the race/class or abilities are undefined
157            let x = this.joinCharacteristics();                       // so that they don't show up on the various menus if they are
158            let y = this.joinAbilities();
159            let description = ''
160            if(x == undefined && y == undefined){
161                description = "Character Name: " + this.selectedCharacter.name + '\n' ;
162            }else if(x == undefined){
163                description = "Character Name: " + this.selectedCharacter.name + '\n' + y + '\n' ;
164            }else if(y == undefined){
165                description = "Character Name: " + this.selectedCharacter.name + '\n' + x + '\n' ;
166            }else{
167                description = "Character Name: " + this.selectedCharacter.name + '\n' + x + '\n' + y + '\n' ;
168            }
169            return description;
170        }
171
172        createCharacter(){                                            // Creates a character based on the name entered by the user
173            let name = prompt(`Enter name for new character: `);
174            this.characters.push(new Character(name));
175        }
176
177        viewCharacteristics(){                                        // Allows the user to see the characteristics menu to then create, change,
178            let index = prompt(`Enter the index of the character you wish to edit:`);  // or delete a character's race/class
179            if (index > -1 && index < this.characters.length){
180                this.selectedCharacter = this.characters[index];
181                var description1 = this.isItUndefined();
182
183                let selection = this.showCharacteristicsMenuOptions(description1);
184                    switch (selection){
185                        case '1':
186                            this.createRaceAndClass();
187                            break;
188                        case '2':
189                            this.changeRaceAndClass();
190                            break;
191                        case '3':
192                            this.deleteRaceAndClass();
```

```
191                            case '3':
192                                this.deleteRaceAndClass();
193
194                        }
195                    }
196                }
197
198        viewAbilities(){                                        // Allows the user to see the abilities menu to then create, change,
199            let index = prompt(`Enter the index of the character you wish to edit:`);    // or delete a character's abilities
200            if (index > -1 && index < this.characters.length){
201                this.selectedCharacter = this.characters[index];
202                var description2 = this.isItUndefined();
203
204                let selection = this.showAbilityMenuOptions(description2);
205                    switch (selection){
206                        case '1':
207                            this.createAbility();
208                            break;
209                        case '2':
210                            this.changeAbility();
211                            break;
212                        case '3':
213                            this.deleteAbility();
214
215                    }
216
217            }
218        }
219
220        viewCharacter(){                                        // Allows the user to see the character menu to choose if they want to
221            let index = prompt(`Enter the index of the character you wish to view:`);     // or edit a character's race/class or abilities
222            if (index > -1 && index < this.characters.length){
223                this.selectedCharacter = this.characters[index];
224                let description = this.isItUndefined();
225
226                let selection = this.showCharacterMenuOptions(description);
227                    switch (selection){
228                        case '1':
229                            this.viewCharacteristics();
230                            break;
231                        case '2':
232                            this.viewAbilities();
233
234                    }
235            }
236        }
237        deleteCharacter(){                                      // Deletes a created character
238            let index = prompt(`Enter the index of the character you wish to delete:`);
239            if (index > -1 && index < this.characters.length) {
240                this.characters.splice(index, 1);
241            }
242        }
243
244        createAbility(){                                        // Creates an ability with a score based on user input
245            let ability = prompt('Enter ability: ');
246            let score = prompt('Enter ability score: ');
247            this.selectedCharacter.abilities.push(new Ability(ability, score));
248        }
249
250        changeAbility(){                                        // Changes the ability score of a chosen ability that's already been created by the user
251            let index = prompt('Enter the index of the ability you wish to change: ');
252            let changedScore = prompt('Enter the new score: ');
253            if (index > -1 &&  index < this.selectedCharacter.abilities.length){
254                this.selectedCharacter.abilities[index].score = changedScore;
255            }
256        }
257
258        deleteAbility(){                                        // Deletes an ability and its score from a character
259            let index = prompt('Enter the index of the ability you wish to delete: ');
260            if (index > -1 &&  index < this.selectedCharacter.abilities.length){
261                this.selectedCharacter.abilities.splice(index, 1);
262            }
263        }
264
265        createRaceAndClass(){                                   // Creates a race and class based on user input, and only allows one to exist at a time
266            if(this.selectedCharacter.characteristics.length === 0){
267            let race = prompt('Enter race: ');
268            let rpgClass = prompt('Enter class: ');
269            this.selectedCharacter.characteristics.push(new Characteristics(race, rpgClass));
270            }else{
271                alert("You can only have one race and class!");
272            }
273        }
274
275        changeRaceAndClass(){                                   // Changes the selected character's race and class based on new user input
276            let changedRace = prompt('Enter new race: ');
277            let changedClass = prompt('Enter new class: ');
278            this.selectedCharacter.characteristics[0].race = changedRace;
279            this.selectedCharacter.characteristics[0].rpgClass = changedClass;
280        }
281
```
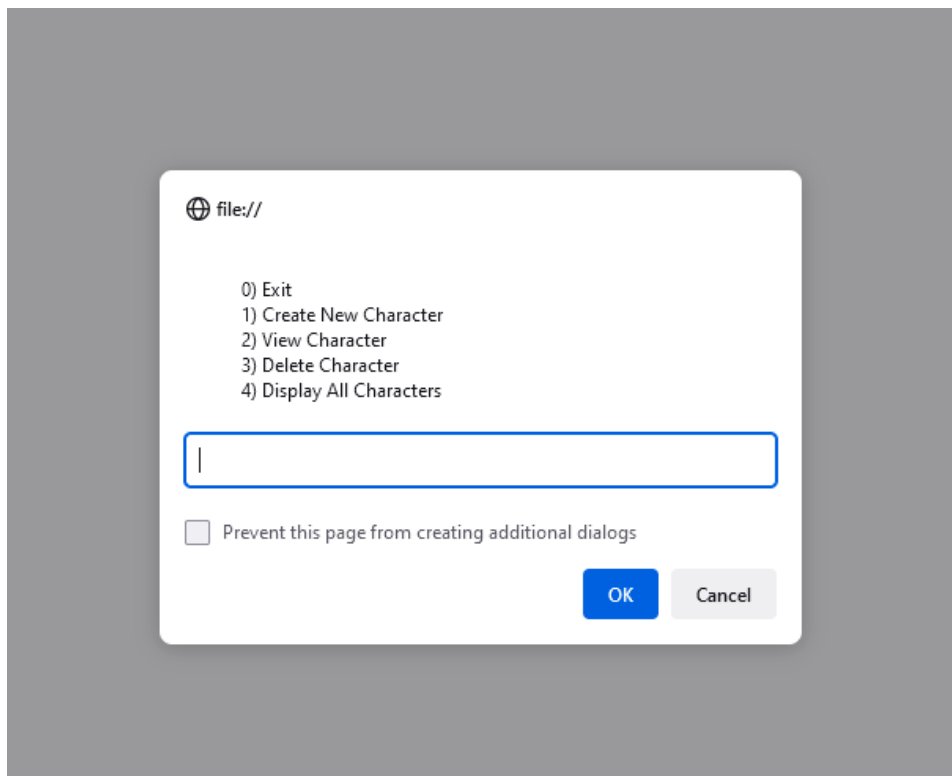
```
281
282      deleteRaceAndClass(){                                    // Deletes the selected character's race and class
283           this.selectedCharacter.characteristics.pop();
284      }
285  }
286
287  let menu = new Menu();
288  menu.start();
289
290
```
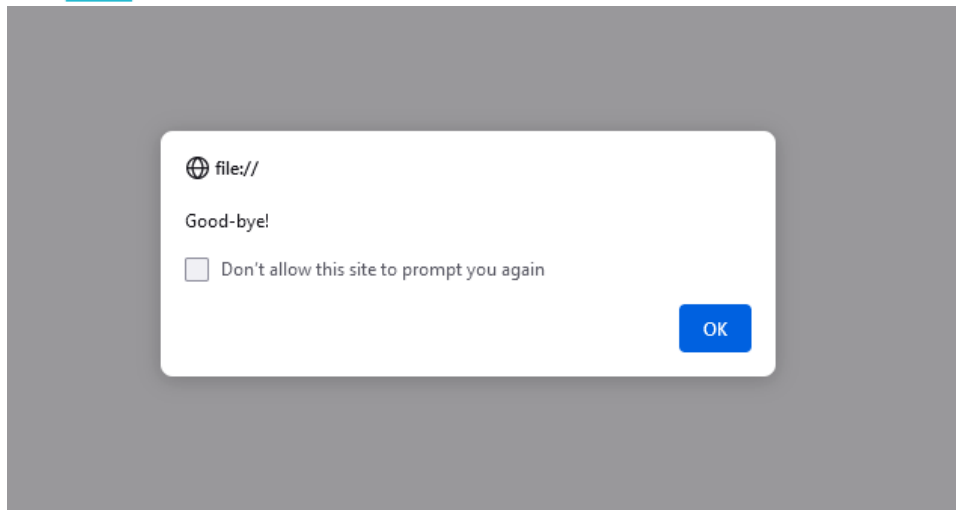
Screenshots of Running Application:

Main Menu:
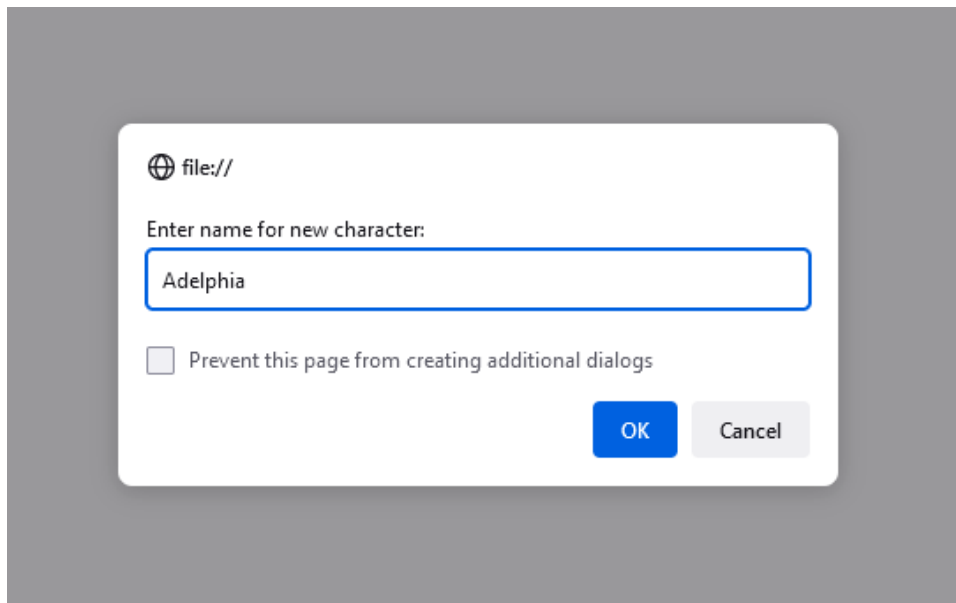

Exit:

Create a New Character:

Choose Character Prompt:

View Character Menu:

Characteristics Menu:

⊕ file://

0) Back
1) Enter Race and Class
2) Change Race and Class
3) Delete Race and Class

// Character Name: Adelphia

☐ Prevent this page from creating additional dialogs

[OK] [Cancel]

Enter Race and Class:

⊕ file://

Enter race:

Tabaxi

☐ Prevent this page from creating additional dialogs

[OK] [Cancel]

Ability Menu:

Create an Ability:

A Character with several abilities and a race and class showing:

⊕ file://

        0) Back
        1) View Characteristics
        2) View Abilities

        Character Name: Adelphia
0) Tabaxi Bard

0) Charisma - 18
1) Strength - 12
2) Constitution - 10

[   ]

☐ Prevent this page from creating additional dialogs

                        OK   Cancel

Changing an ability score:

⊕ file://

Enter the index of the ability you wish to change:

[   ]

☐ Prevent this page from creating additional dialogs

                        OK   Cancel

file://

Enter the new score:

15

☐ Prevent this page from creating additional dialogs

OK   Cancel

file://

```
        0) Back
        1) View Characteristics
        2) View Abilities
        ----------------------
        Character Name: Adelphia
0) Tabaxi Bard

0) Charisma - 18
1) Strength - 12
2) Constitution - 15
```

☐ Prevent this page from creating additional dialogs

OK   Cancel

After deleting an ability:

Only one race/class allowed warning:

Change Race and Class:

```
⊕ file://

        0) Back
        1) View Characteristics
        2) View Abilities
        ----------------------
        Character Name: Adelphia
0) Satyr Paladin

0) Charisma - 18
1) Constitution - 15


[                                                    ]

☐ Prevent this page from creating additional dialogs

                                    [ OK ]  [ Cancel ]
```
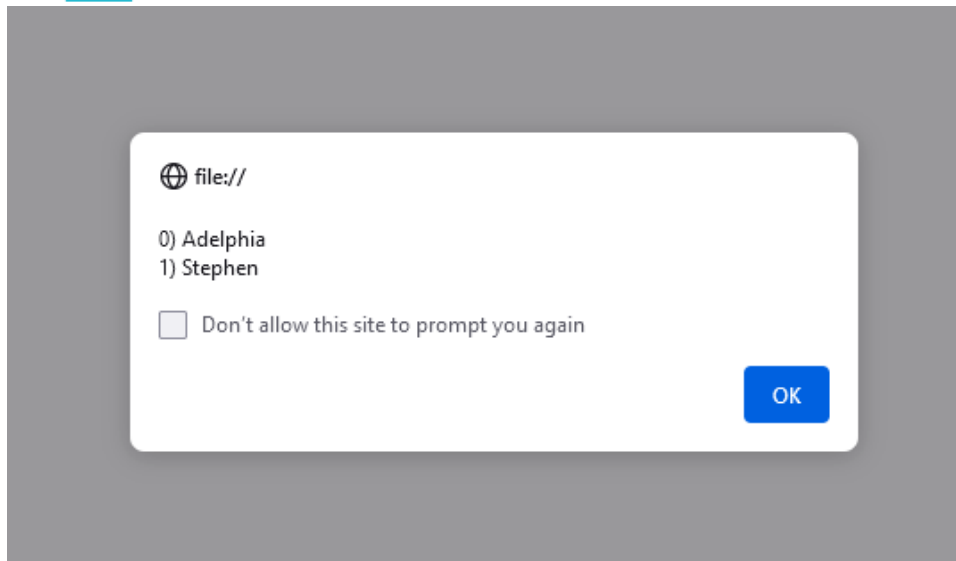
After deleting race and class:

Display Characters:

Delete Character

URL to GitHub Repository:

https://github.com/aheiser2/Week-5-Menu-App.git