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.
   1. Use at least one array.
   2. Use at least two classes.
   3. Your menu should have the options to create, view, and delete elements.

**Screenshots of Code:**

class FoodItem {

    constructor(name, price) {

      this.name = name;

      this.price = price;

    }

    describe() {

      return `${this.name} costs ${this.price}`;

    }

}

class ShoppingCart {

  constructor(name){

    this.name = name;

    this.foodItems = [];

  }

  describe() {

    return `${this.name} has ${this.foodItems.length} food items`;

  }

}

class Menu {

  constructor() {

    this.foodItems = [];

    this.selectedFoodItem = null;

  }

  start() {

    let selectedMenuOption = this.showMainMenuOptions();

    while (selectedMenuOption != 0) {

      switch (selectedMenuOption) {

        case '1':

          this.createItem();

          break;

        case '2':

          this.deleteItemByIndex();

          break;

        case '3':

          this.displayCart();

          break;

        case '4':

          this.totalOfItems();

          break;

        default:

            selectedMenuOption = 0;

      }

      selectedMenuOption = this.showMainMenuOptions();

    }

    alert(`Please come again!`);

  }

  showMainMenuOptions() {

    return prompt(`

      0) Exit

      1) Enter item

      2) Remove item

      3) View shopping cart

    `);

  }

  createItem() {

    let name = prompt(`Enter name for new grocery item selected:`);

    let price = prompt(`Enter price for ${name}`);

    this.foodItems.push(new FoodItem(name, price));

  }

  displayCart() {

    let cartString = '';

    for (let i = 0; i < this.foodItems.length; i++) {

      cartString += i + ') ' + this.foodItems[i].name + ' ' + this.foodItems[i].price + '\n';

    }

    alert(cartString);

  }

  deleteItem(index) {

    if (index > -1 && index < this.foodItems.length) {

      this.foodItems.splice(index, 1);

    }

  }

  deleteItemByIndex() {

    let index = prompt(`Enter the index of the player you wish to delete:`);

    if (index > -1 && index < this.foodItems.length) {

      this.foodItems.splice(index, 1);

    }

  }

}

let menu = new Menu();

menu.start();

**Screenshots of Running Application:**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**URL to GitHub Repository:**