Week 5: Coding Assignment

URL to GitHub Repository:

https://github.com/soccermoe14/promineo-week5

URL to Your Coding Assignment Video:

https://youtu.be/US5MkThgMDE

Instructions:

- In Visual Studio Code, write the code that accomplishes the objectives listed below and ensures that the code compiles and runs as directed.
- Create a new repository on GitHub for this week's assignments and push this document, with your project code, to the repository.
- Include the URLs for this week's repository and video where instructed.
- Submit this document as a .PDF file in the LMS.

Coding Steps:

- 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:
    - Use at least one array.
    - Use at least two classes.
    - Your menu should have the options to **create**, **view**, and **delete** elements.

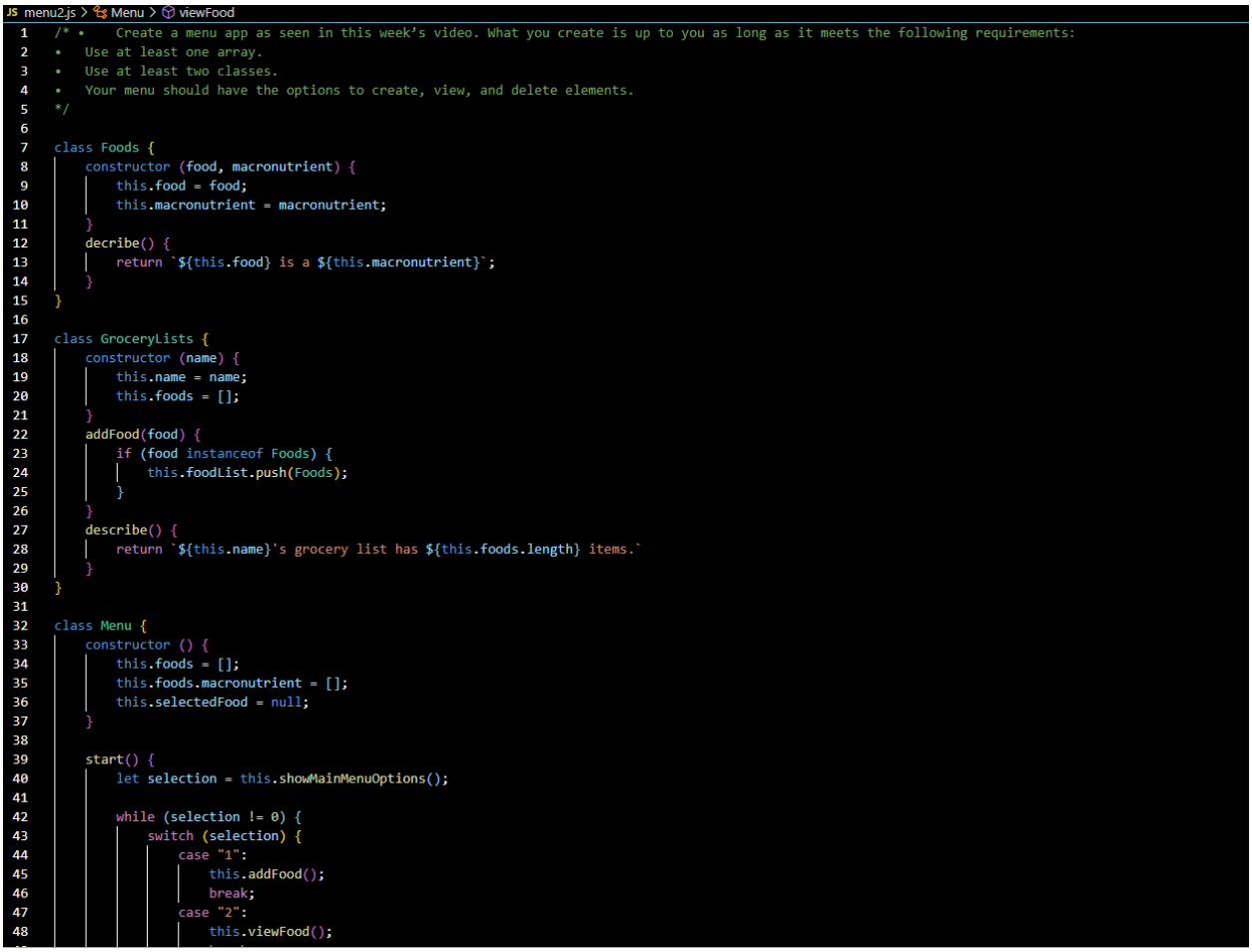
Video Steps:

- Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed.
- This video should be done using screen share and voice over.
- This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend.
    - You can create a new meeting, start screen sharing, and start recording.
    - This will create a video recording on your computer.
- This should then be uploaded to a publicly accessible site, such as YouTube.
    - Ensure the link you share is **PUBLIC** or **UNLISTED**!

Week 5: Coding Assignment

- If it is not accessible by your grader, your project will be graded based on what they can access.

Screen shots of the code that somewhat works (menu2.js):



```
JS menu2.js > Menu > viewFood
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:
2  •   Use at least one array.
3  •   Use at least two classes.
4  •   Your menu should have the options to create, view, and delete elements.
5  */
6
7  class Foods {
8      constructor (food, macronutrient) {
9          this.food = food;
10         this.macronutrient = macronutrient;
11     }
12     decribe() {
13         return `${this.food} is a ${this.macronutrient}`;
14     }
15 }
16
17 class GroceryLists {
18     constructor (name) {
19         this.name = name;
20         this.foods = [];
21     }
22     addFood(food) {
23         if (food instanceof Foods) {
24             this.foodList.push(Foods);
25         }
26     }
27     describe() {
28         return `${this.name}'s grocery list has ${this.foods.length} items.`
29     }
30 }
31
32 class Menu {
33     constructor () {
34         this.foods = [];
35         this.foods.macronutrient = [];
36         this.selectedFood = null;
37     }
38
39     start() {
40         let selection = this.showMainMenuOptions();
41
42         while (selection != 0) {
43             switch (selection) {
44                 case "1":
45                     this.addFood();
46                     break;
47                 case "2":
48                     this.viewFood();
```

PROMINEO TECH

Week 5: Coding Assignment

```
                    break;
                case "2":
                    this.viewFood();
                    break;
                case "3":
                    this.deleteFood();
                    break;
                case "4":
                    this.displayFoods();
                    break;
                default:
                    selection = 0;
            }
            selection = this.showMainMenuOptions();
        }
        alert ("No food for you!");
    }
    showMainMenuOptions() {
        return prompt (`
        0) Exit
        1) Add Food
        2) View Food
        3) Delete Food
        4) Displacy All Foods
        `);
    }
    showGroceryListMenuOptions(food) {
        return prompt (`
        0) Back
        1) Add Grocery List
        2) Remove Grocery List

        `);
    }
    displayFoods() {
        let foodString = " ";
        for (let i = 0; i < this.foods.length; i++) {
            foodString += i + ") " + this.foods[i].food + "\n";
        }
        alert(foodString);
    }
    addFood() {
        let food = prompt ("Add food");
        let macronutrient = prompt ("Add macronutrient");
        this.foods.push(new Foods(food, macronutrient));
    }
    viewFood() {
        let index = prompt("Enter index number of food");
```

PROMINEO TECH

Week 5: Coding Assignment

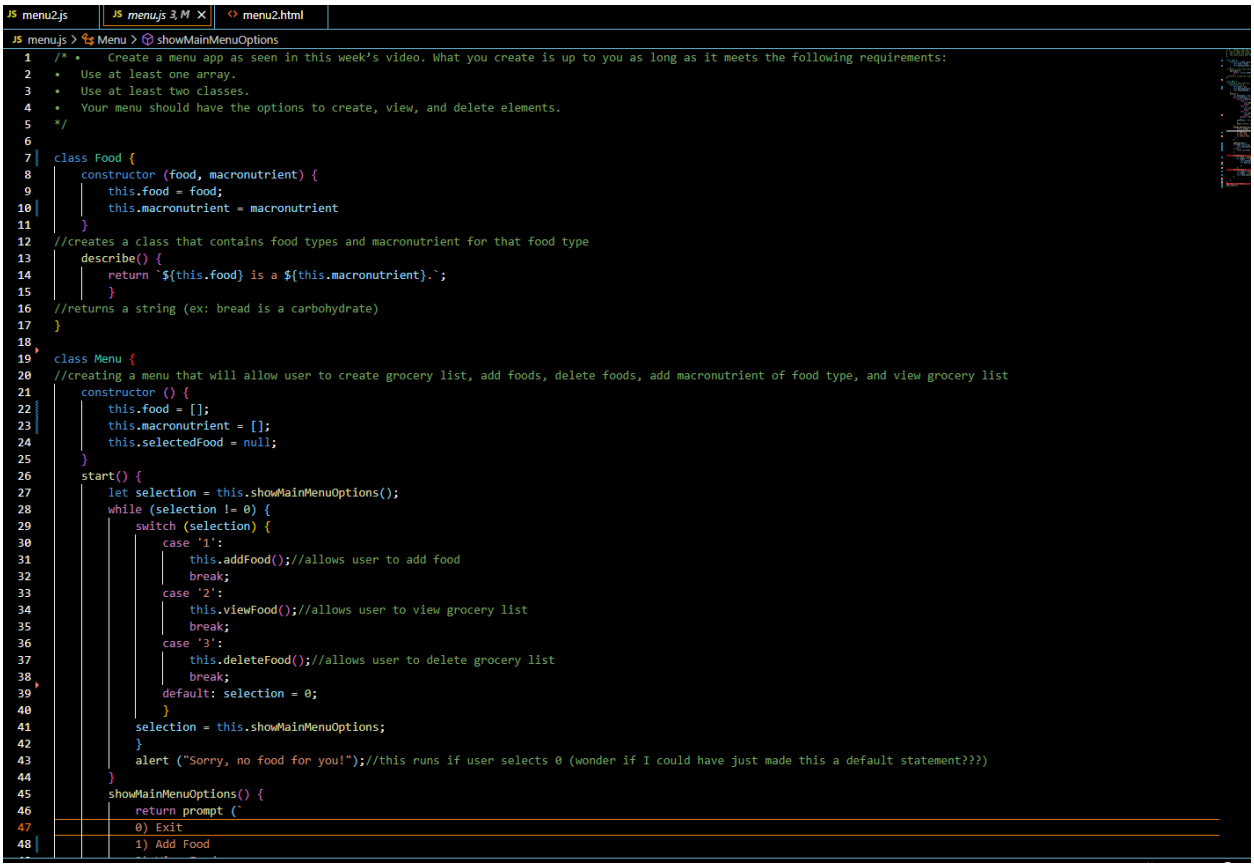
menu2.js menu2.html

menu2.js > ...

```
88          let food = prompt ("Add food");
89          let macronutrient = prompt ("Add macronutrient");
90          this.foods.push(new Foods(food, macronutrient));
91      }
92      viewFood() {
93          let index = prompt("Enter index number of food");
94          if (index > -1 && index < this.foods.length) {
95              this.selectedFood = this.foods[index];
96              let description = `${this.selectedFood}`;
97              let selection = this.showGroceryListMenuOptions(description);
98              switch (selection) {
99                  case "1":
100                     this.addGroceryList();
101                     break;
102                 case "2":
103                     this.deleteGroceryList();
104             }
105         }
106     }
107     deleteFood() {
108         let index = prompt ("Enter index of food to delete");
109         if (index > -1 && index < this.foods.length) {
110             this.foods.splice(index, 1);
111         }
112     }
113     addGroceryList() {
114         let name = prompt("Name of grocery list");
115         this.selectedFood.groceryList.push(new GroceryLists(name));
116     }
117     deleteGroceryList() {
118         let index = prompt ("Enter index of grocery list to remove");
119         if (index > -1 && index < this.selectedFood.GroceryLists.length) {
120             this.selectedFood.GroceryLists.splice(index, 1);
121         }
122     }
123 }
124
125 let menu = new Menu();
126 menu.start();
127
```

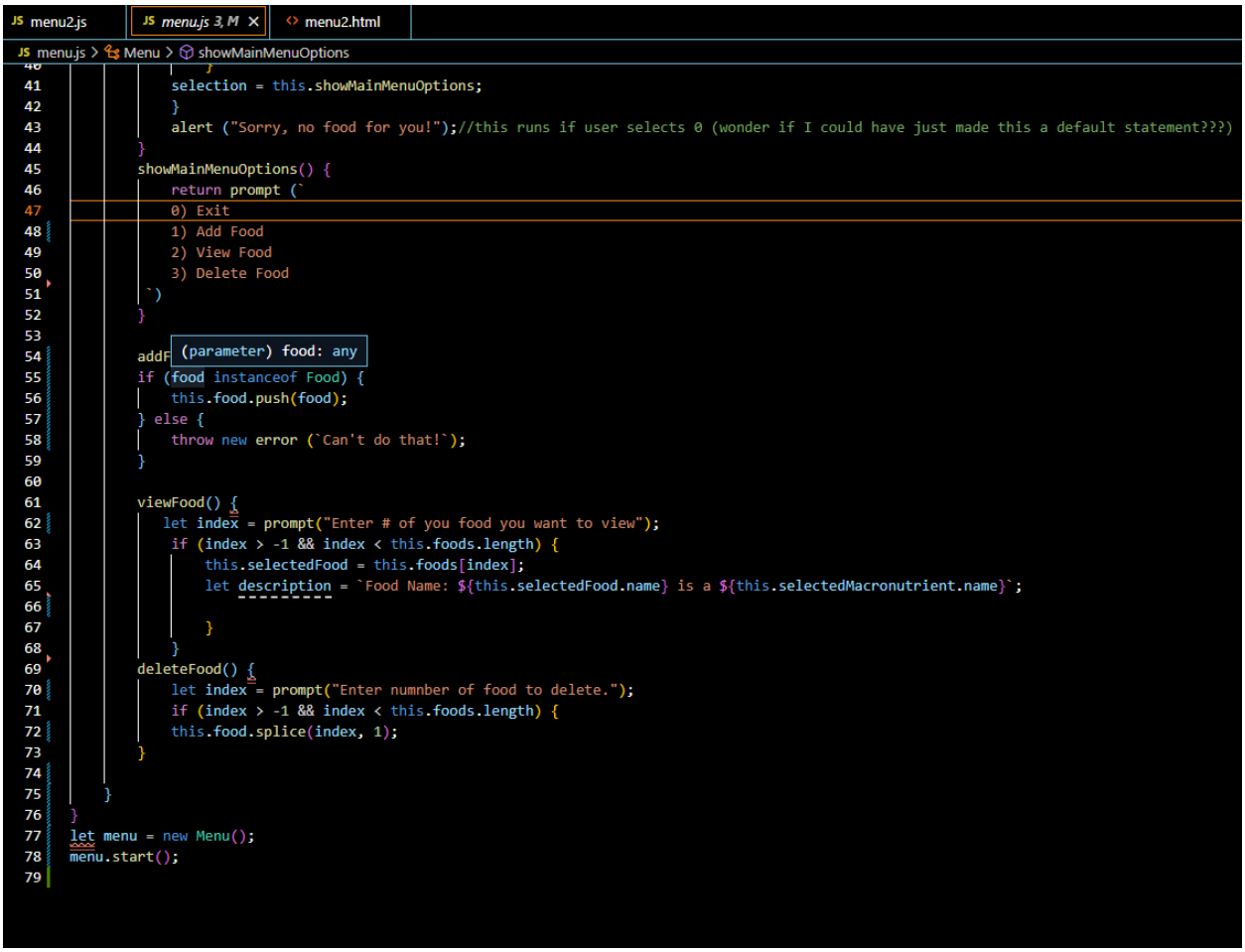
This was my first attempt, but I've removed a lot of code as I was testing and getting errors. That mish mash of code is on the screen shots from the word document below as well. This code also include notes; however, like I said on my video, it seemed like the notes were causing errors, so I removed them.

Week 5: Coding Assignment



PROMINEO TECH

Week 5: Coding Assignment



Week 5: Coding Assignment

```
addGroceryList() {
        let name = prompt("Enter name of grocery list.");
        this.selectedFood.groceryList.push(new GroceryList(name));
    }


 showGroceryMenuOptions(foodInfo) {
        return prompt (`
        0) Back
        1) Add Grocery List
        2) Delete Grocery List
        ----------
        ${foodInfo}
        `)
    }


class PersonalList {
   constructor (name, food) {
      this.name = name;
      this.foods = [];
   }
//creates a class with names so we can have grocery lists for different people
   addFood(food) {
      if (food instanceof Foods) {
         this.groceryList.push(Foods);
      } else {
```

Week 5: Coding Assignment

```
        throw new Error("Nope, can't do that!");
      }
```

```
//adds a new food type to our list
  }
  describe() {
    return `${this.name}'s grocery list has ${this.groceryList.length} items.`;
  }
//returns a string (ex: Marin's grocery list has 10 items)
}
```

```
removeGroceryList() {
      let index = prompt("Enter number of grocery list to remove.");
      if (index > -1 && < this.selectedFood.groceryLists.length) {
        this.selectedFood.groceryLists.splice(index, 1);
      }
    }
  }
```

Week 5: Coding Assignment

```
let menu = new Menu();

menu.start();
```

```
displayGroceryLists() {
        let groceryListString = ' ';
        for (let i = 0; i , this.groceryList.length; i++) {
          groceryListString += i + ') ' + this.groceryLists[i].food + '\n';
```

```
        }//for loop that will iterate through the grocery lists and number them (double check that I am stating this correctly)
          alert(groceryListString);
      }
      createGroceryList() {//creates different grocery lists
        let name = prompt ('Enter name for this grocery list:');
        this.groceryLists.push(new GroceryList(name));//pass in name from prompt into the new grocery list. That name will be pushed to grocery lists array.
      }
      viewGroceryList() {//creates ability to view grocery lists
        let index = prompt ('Enter the index of the grocery list you wish to view:');
```

Week 5: Coding Assignment

```
grocery list. That name will be pushed to grocery lists array.
        }
        viewGroceryList() {//creates ability to view grocery lists
          let index = prompt ('Enter the index of the grocery list you wish to view:');
          if (index > -1 && index < this.groceryLists.length) {//validates user input so we don't get an
error if input is < 0 or > grocery list array
            this.selectedGroceryList = this.groceryLists[index];
            let description = 'This is' + this.selectedGroceryList + '\n';

            for (let i = 0; i < this.selectedGroceryList.foods.length; i++) {
              description += i + ') ' + this.selectedGroceryList.foods[i].name + ' - '
                + this.this.selectedGroceryList.foods[i],position + '\n';//this will build the list of foods on
the grocery list
            }
          let selection = this.showGroceryMenuOptions(description);//this still need to be built. Will pass
in description of
          //grocery list to show grocery menu options and implement showGroceryMenu OPtions to
diplay the groceries
            switch (selection) {
              case '1':
                this.createFood();
                break;
              case '2':
                this.deleteFood();
```

Week 5: Coding Assignment

```
                }


            }
        }
}
let menu = new Menu();
menu.start();


//NEED TO GO BACK ANS ADD MY SHOW GROCERY MENU OPTIONS WHERE I CAN ADD FOOD ITEMS
```