Lesson Plan: Introduction to Arrays and Inventory Management

Grade Level: Middle School

Time estimate: 2 (60 minute) periods

Overview: This two day lesson plan introduces students to arrays. The assignment uses the “building a lunchbox” theme to help students understand how arrays can be used to store and manage a collection of items or “inventory”. Students will edit and extend a provided code template used to create a “LunchBox” class that can add, remove, and list items. This assignment provides a basis for more data structures in the future.

Day 1:

- Objective: What is an array?
  - Implement a class with a constructor and fields
  - Use the provided code to add items to an array
- Materials
  - VS Code
  - Github
  - Computer
  - Post-It Notes 3x student
  - LunchBox.js

```
class LunchBox {
  constructor(name) {
    this.name = name;
    this.inventory = [];
  }

  addItem(item) {
    // Put your code here to add an item to the inventory
  }

  removeItem(item) {
    // Put your code here to remove an item to the inventory
  }

  listItem() {
    // Put your code here to print all items in the inventory
  }
}
```

- Instruction
  - Say: “Imagine you have to pack your lunchbox tomorrow, what items would you put in your lunchbox? Turn and talk.”
  - Call students up to write their ideas on the board
  - Inventory = List of things someone has
    - Array is list that can hold diff items like lunchbox holds food
  - Give students code “LunchBox” (about 20 mins)
  - Explain parts like “class”, “constructor” and “inventory” fields
  - “addItem” function adds an item to the end of the inventory array using “.push()”

    - Students build lunchbox class (25 mins)
        - Students use template, create new “LunchBox” object and use “addItem” function to add 3 food items of their choice
    - Check for understanding, and help students during this time.
- Ending (5 mins)
    - Ask students to tell partner what an array is and how “addItem” function works
    - Say: Tomorrow we will learn about how to remove and list the items in your LunchBox

Day 2:

- Objectives
    1. Students will be able to change arrays by removing elements with “splice()”
    2. Students will be able to list elements in an array using a loop
    3. Students will be able to provide specific and respectful feedback
- Prior Knowledge:
    1. Day 1
    2. For loops
- Materials
    1. Dominos x 10 for visual
    2. VS Code
    3. Computer
    4. Post-it notes (3x student)
- Instruction
    1. Recall, what is an array? What does the addItem function do?
        - Introduce removeItem function as way to take things out of inventory
    2. Explain splice() method to remove item from a specific position in the array
        - Line up dominos into array and pick up domino in middle to demonstrate slice
    3. listItem function will use for loop to print the inventory array
    4. Demonstrate an example of full code with an apple, sandwich, and cookie, and then remove the apple. End by listing out items. This gives students an example of what the code is actually supposed to do and scaffolds the code for them.
    5. Tell students they now implement removeItem and listItem functions in their LunchBox class.
        - Create LunchBox and add 4 items that the teacher hasn’t used
        - Print initial inventory by using listItem function
        - Remove one item from the lunchbox
        - Add a new item to the lunchbox
        - Use the listItem function to print the inventory
    6. Gallery walk
        - Students instructed to leave projects open on their computers.
        - Explain the feedback process
            - Every student write name on 3 postits
            - Write 2 things classmate did well and 1 thing to improve
            - IT IS IMPORTANT TO BE SPECIFIC AND RESPECTFUL
        - Supervise the students while they are walking around

Name: Date: Due:

# Assignment: Lunchbox Manager

**Overview:** Welcome new lunchbox manager! You have been hired to create a digital LunchBox that can manage a list of food items. We will call your LunchBox a class because it says what sort of data can be put in it, in this case delicious food of your choice manager. You have a code template to start "LunchBox.js". Your final code should be able to add, remove, and list items in the lunchbox inventory.

**Instructions:**

1. In the code complete the "addItem()", "removeItem()", and "listItem()".
2. Create a new "LunchBox object and give it a name ex: myLunchbox
3. Add at least 4 items to your lunchbox. You must choose different items than the teacher..
4. Use the "listItem()" method to see everything you have in the lunch box.
5. Remove one of the items.
6. Add new items to your lunchbox.
7. Use the listItem() method one last time to show your boss what is in your lunchbox.

***Resumen:*** *¡Bienvenido, nuevo administrador de loncheras! Te han contratado para crear una lonchera digital que pueda gestionar una lista de alimentos. Llamaremos a tu lonchera una clase porque indica qué tipo de datos se pueden incluir en ella; en este caso, un administrador de deliciosa comida a tu elección. Tienes una plantilla de código para iniciar "LunchBox.js". Tu código final debería poder añadir, eliminar y listar artículos en el inventario de la lonchera.*

***Instrucciones:***

1. *En el código, completa "addItem()", "removeItem()" y "listItem()".*
2. *Crea un nuevo objeto "LunchBox" y asígnale un nombre, por ejemplo: myLunchbox.*
3. *Añade al menos 4 artículos a tu lonchera. Debes elegir artículos diferentes a los del profesor.*
4. *Usa el método "listItem()" para ver todo lo que hay en la lonchera.*
5. *Elimina uno de los artículos.*
6. *Añade un nuevo artículo a tu lonchera.*
7. *Usa el método "listItem()" por última vez para mostrarle a tu jefe qué hay en tu lonchera.*

Rubric:

| Criteria | 5 - Exceeds Expectations | 4 - Meets Expectations | 3 - Approaching Expectations | 2 - Needs Improvement |
|---|---|---|---|---|
| Understanding of Arrays | Demonstrates a thorough understanding of arrays by correctly adding, removing, and listing items with no errors. | Demonstrates good understanding of arrays by correctly adding, removing, and listing items with minor errors. | Demonstrates partial understanding of arrays but makes several errors in adding, removing, or listing items. | Shows minimal or incorrect understanding of how to use arrays. |
| Application of Coding Concepts | Correctly applies all required concepts (class, constructor, | .push(), .splice(), for loop) with a fully functional LunchBox class. | Applies most concepts with minor mistakes that do not significantly affect the expected functionality. | Attempts to apply coding concepts but with errors or missing concepts affecting at least one of the expected functionalities. |
| Project Completion | Creates a | LunchBox with inventory, successfully adds/removes/lists items, and prints accurate inventory lists at each step. | Creates a | LunchBox with inventory, successfully adds and removes items, and prints an accurate list at least once. |
| Peer Feedback (Day 2) | Provides 2 specific positive comments and 1 specific area for improvement using clear, respectful, and thoughtful language. | Provides 2 positive comments and 1 area for improvement that are specific and respectful. | Provides feedback with limited specificity or some lack of respectful tone. | Feedback is missing, very disrespectful, or not relevant to the project. |

Satisfactory Code

```
class LunchBox {
  constructor(name) {
    this.name = name;
    this.inventory = [];
  }

  addItem(item) {
    this.inventory.push(item);
    console.log(`${item} has been added to ${this.name}'s lunchbox.`);
  }

  removeItem(item) {
    const itemIndex = this.inventory.indexOf(item);
    if (itemIndex > -1) {
      this.inventory.splice(itemIndex, 1);
      console.log(`${item} has been removed from ${this.name}'s lunchbox.`);
    } else {
      console.log(`${item} is not in the lunchbox.`);
    }
  }

  listItem() {
    console.log(`${this.name}'s Lunchbox Inventory:`);
    for (let i = 0; i < this.inventory.length; i++) {
      console.log(`- ${this.inventory[i]}`);
    }
  }
}

let myLunchbox = new LunchBox("Maria's");
myLunchbox.addItem("sandwich");
myLunchbox.addItem("apple");
myLunchbox.addItem("cookie");
myLunchbox.addItem("carrots");

myLunchbox.listItem();

myLunchbox.removeItem("apple");
myLunchbox.addItem("juice box");

myLunchbox.listItem();
```

This is a great example of code that would receive the full 20 points based on my rubric. The simulation of Maria's lunchbox runs and shows all 4 items, removing the apple and adding juice box. It also lets the user know if the item is not in the lunchbox. It correctly uses the addItem using .push(). My code is functional and fulfills all program requirements.

Unsatisfactory Code:

```
1  class LunchBox {
2    constructor(name) {
3      this.name = name;
4      this.inventory = [];
5    }
6    addItem(item) {
7      this.inventory.push(item);
8      console.log(`added ${item} to ${this.name}'s lunchbox`);
9    }
10   removeItem(item) {
11     console.log(`${item} has been removed from ${this.name}'s inventory`);
12     this.inventory.splice(this.inventory.indexOf(item), 1);
13   }
14   listItem() {
15     console.log(`${this.name}'s inventory`);
16     for (let i = 0; i <= this.inventory.length; i++) {
17       console.log(`- ${this.inventory[i]}`);
18     }
19   }
20 }
21
22 let myLunchbox = new LunchBox("Alex's");
23 myLunchbox.addItem("burger");
24 myLunchbox.removeItem("fries");
25 myLunchbox.listItem();
```

This code is a poor example because it might produce errors or unexpected outputs. Students might confuse similar sounding functions like in this example where it uses .pull() instead of .push(). There is a mistake on Line 16 that is an error in the loop condition.

**Reflection:**

This feels like how I would ideally teach an algorithms and data structures class. I think Data Structures and Algorithms was one of the most challenging classes for us to learn. If I were to take this class again, I would want it to be laid out the way I've demonstrated here where it has clear goals that can be directly posted and outlined in different agile goals. I think the importance of stand-ups and sprints cannot be understated enough. One of the challenges for me was learning how to code with a partner. I really struggled with understanding what it was like to balance out the workload and communicating needs. I think it would be more helpful having a 3rd team member who could delegate the work and organize us better. This class ideally would have included more coding and scaffolding for the students especially because it was last minute made into an asynchronous course.