Sophia Qin

PUI Assignment 6B

5 Programming Concepts

1) Using document.getElementByID and document.getElementByClass is helpful in pointing to and changing a DOM's element's content. By using a class or ID as a unique identifier for each item, item objects can be added, updated, and removed from the cart easily because it is reusable. One example is updating the shopping cart badge by calling document.getElementByID('shopping-cart-badge') and updating the number or data count inside the notification by looking at the cart length.

```
function updateBadge() {

    let badge = document.getElementById('shopping-cart-badge');

    let notification = document.getElementById('notification');

    if (cart.length) {

        badge.setAttribute('data-count', cart.length);

        notification.style.display = 'inline-block';

    } else {

        notification.style.display = 'none';

    }

}
```

2) Using event-driven programming such as event listeners and event-handlers is helpful in executing mouse commands from the user to get state updates and to update the cart. This is helpful because users need to input commands through keyboard or mouse and the application needs to put it in an event queue, so it can be dispatched. One example of this is when removing the items in the cart. I had an event listener to identify when the user clicks the "Close" button, and then remove it from the cart.

```
 for (let i = 0; i < removeCartItemsButtons.length; i++) {

        const closeButton = removeCartItemsButtons[i];

        closeButton.addEventListener('click', () =>
removeFromCart(i));

    }
```

3) Grouping information into objects is helpful in the cart as you want to display multiple items inside the cart with the same characteristics (frosting type and quantity) which can

be customized by the user. When you want the cart items to be added and be updated with the quantity and frosting type you choose, you can store that information inside an object, which can be changed by the specified variables. Making new objects can be done through constructors. One example of this is making caramel pecan an object, and then putting frosting type, quantity, and total price (quantity * priceitem) to be displayed in the cart.

```
cart.push({
        name: 'Carmel Pecan',
        frosting: (frostingType) ? frostingType.innerText :
'None',
        quantity: quantity,
        price: 9.99,
        totalPrice: 9.99 * quantity
    });
```

4) Using storage such as localStorage to save objects, and JSON.parse to retrieve items, is helpful in storing objects in the web. For example, when attempting to store things in the cart, you can write JSON.parse(localStorage) to retrieve the array of items that were stored in the cart and display it.

```
let cart = (localStorage.getItem('cart'))
    ? JSON.parse(localStorage.getItem('cart'))
    : [];
updateBadge();
```

5) Arrays are useful in storing a fixed size collection of elements, (such as frosting types) and accessing them to display the user's choice in the cart item. An example of this is when we create an array from the frosting options and the program returns or displays the active button for the frosting type that the user clicks into the cart.

```
let frostingButtons =
document.getElementById('frostings');
let frostingType =
Array.from(frostingButtons.children).find(button => {
        return button.classList.contains('active');
```

What challenges or bugs did you encounter? How did you overcome these challenges?

Sophia Qin

There was a challenge in trying to delete delete things in an array of elements one by one, as I needed to remember arrays start from 0. This was in the context of trying to delete things from the cart. Initially, when sequentially deleting elements in the array, everything got shifted because I was calling elements by their corresponding position. To overcome this challenge, I needed recompute the indexes differently and call/delete “0” position each time instead. I have to make sure to remember arrays start from 0 and take that in consideration when calling elements inside of it. Another challenge was not knowing where to start, for example, when trying to update cart and total. Searching on google was good to get an idea on how to start. I found from online that you could have for loop and cycle through all the items in an array and add the item’s price to the total.