Maddy Cha
Section A
mecha@andrew.cmu.edu

## Homework 6B Reflection

I started off by using JavaScript to update and store information from the product page. Similarly to how we worked in lab, I created an object that stores yarn and material information for each pillow being added to the cart. I then stored each of these objects in an array that represented the cart. I had a bit of trouble because I wanted to have the add to cart button grayed out if the yarn and material choice had not been made yet, but I solved it by checking whether or not they were using an if statement. I checked to make sure both the yarn and material choice had values (using !== undefined) and then changed the color of the button to demonstrate functionality.

Once I had the style correct, I realized it was still possible to add to cart without a yarn choice (yarn choice would show up as “null” on the cart page). To fix this, I created a variable that would store whether or not the yarn choice and material choice had been made (although as material choice was defaulted to the first choice, it was always stored). Once a yarn choice was made, the variable would be set to true, and the item could be added to the cart.

I had a couple of issues when coding the functionality of my shopping cart. For one, it was difficult for me to figure out how to display all of the cart information without overriding each item. I used text nodes and innerHTML to update each and ran it through a for loop based on the length of the cart array so that it would create new elements for each of the items in the array. Another issue I had was that the cart would not save when going to different pages. I had to use localStorage to keep the cart and cart length updated through each of the pages. I made sure to introduce these variables to each of the pages using an onload listener, so that it would be generated when the page/body loaded.

Finally, I had to implement deleting items from the cart. I decided to go about it by splicing the array using an onclick function, but had trouble figuring out which item should be deleted from the cart. I went through a few ideas, and eventually got it to work by setting a value to each of the delete elements and utilizing it as a parameter for my remove from cart function.

## Programming Concepts

1. Objects
    a. I decided to use objects to take in information from the product detail page. My object, pillowChoice, collects the yarn and material information that the user inputs on the page. It also checks to make sure that both choices have been made so that the user can properly add to cart and/or wishlist.

2. Arrays
    a. I stored all of my pillowChoice objects into arrays—one for the cart and one for the wishlist. By storing the objects into the array, I was able to display them and also give the user the option to remove specific items from their cart based on their index.

3. For Loops
    a. I used for loops to populate the information in my cart and wishlist. I set the loop to run for the length of the wishlist/cart arrays (for(var i=0; i < array.length; i++)) and populated the text nodes in the HTML using "i" as index for the array.

4. Local Storage
    a. I stored the cart and wishlist arrays locally so that each of the pages could access them. Even though these arrays are populated on the product detail pages, I saved them locally so that they could be called by the cart/wishlist page and used to populate that content.

5. Event Handlers
    a. I used onclick and onload events to trigger my javascript functions. For example, I had an onclick event on my add to cart button that would trigger add a new object to my array with the pillow and yarn choice made by the user. I also created an onload event on all my pages that would update the number label on the cart based on how many objects were in my cart array.