**Module 6: Portfolio Milestone**

**Github: https://github.com/damansandhu/CSU\_Global\_Module\_6\_Portfolio\_Milestone**

**Pseudocode**

1. **Class ShoppingCart**:
   * **Attributes**:
     + customer\_name: A string initialized to "none" by default.
     + current\_date: A string initialized to "January 1, 2020" by default.
     + cart\_items: A list to store items in the cart.
   * **Methods**:
     + **\_\_init\_\_(customer\_name, current\_date)**:
       - Initialize customer\_name, current\_date, and cart\_items.
     + **add\_item(ItemToPurchase)**:
       - Append ItemToPurchase to cart\_items.
     + **remove\_item(item\_name)**:
       - Find and remove the item by item\_name from cart\_items.
       - If not found, print "Item not found in cart. Nothing removed."
     + **modify\_item(ItemToPurchase)**:
       - Find the item in cart\_items by name.
       - If found, update its description, price, and/or quantity.
       - If not found, print "Item not found in cart. Nothing modified."
     + **get\_num\_items\_in\_cart()**:
       - Return the sum of quantities of all items in cart\_items.
     + **get\_cost\_of\_cart()**:
       - Calculate and return the total cost of all items in cart\_items.
     + **print\_total()**:
       - Print the total number of items and total cost.
       - If the cart is empty, print "SHOPPING CART IS EMPTY."
     + **print\_descriptions()**:
       - Print the descriptions of all items in cart\_items.
2. **Function print\_menu(cart)**:
   * **Loop** until the user chooses to quit (q):
     + Display the menu options.
     + Get the user's choice.
     + Depending on the choice, call the appropriate method from ShoppingCart:
       - a: Add item to the cart.
       - r: Remove item from the cart.
       - c: Change item quantity.
       - i: Output items' descriptions.
       - o: Output shopping cart.
       - q: Quit.

**Explanation**

1. **ItemToPurchase Class**: This class represents an item that can be added to the shopping cart. It has attributes for name, description, price, and quantity.
2. **ShoppingCart Class**:
   * **add\_item()**: Adds an ItemToPurchase object to the cart.
   * **remove\_item()**: Removes an item from the cart by name, with an error message if not found.
   * **modify\_item()**: Modifies an item’s details if found, with an error message if not found.
   * **get\_num\_items\_in\_cart()**: Returns the total quantity of all items.
   * **get\_cost\_of\_cart()**: Returns the total cost of all items.
   * **print\_total()**: Outputs the total number of items and the total cost, with a special message if the cart is empty.
   * **print\_descriptions()**: Outputs the descriptions of all items.
3. **print\_menu() Function**: This function displays a menu, allowing the user to interact with the shopping cart by adding, removing, modifying items, or viewing the cart's contents.

**Step 4: Build the ShoppingCart Class**

* **Constructor**:
  + The ShoppingCart class constructor takes customer\_name and current\_date as parameters, initializing them appropriately, and sets up an empty cart\_items list.
* **Attributes**:
  + customer\_name, current\_date, and cart\_items are all properly initialized and used throughout the class.
* **Methods**:
  + **add\_item()**: Adds an item to cart\_items.
  + **remove\_item()**: Removes an item by name from cart\_items, with a check for the existence of the item.
  + **modify\_item()**: Modifies an existing item's attributes, checking if the item exists and only updating the attributes that are not set to default values.
  + **get\_num\_items\_in\_cart()**: Returns the total quantity of items in the cart.
  + **get\_cost\_of\_cart()**: Calculates and returns the total cost of all items in the cart.
  + **print\_total()**: Outputs the total number of items and their total cost, with a special message if the cart is empty.
  + **print\_descriptions()**: Outputs the description of each item in the cart.

**Step 5: Implement the print\_menu() Function**

* **Menu Functionality**:
  + The print\_menu() function provides a menu for the user to interact with the shopping cart, including options to add, remove, modify, and display items, as well as quit the program. It also includes input validation to ensure the user selects a valid option.

**Step 6: Implement Output Shopping Cart and Item Descriptions**

* **Output Shopping Cart**:
  + Implemented within print\_total(), which displays the total items and their costs, along with a special message if the cart is empty.
* **Output Item Descriptions**:
  + Implemented within print\_descriptions(), which prints each item’s description.

**Additional Requirements**

* **Error Handling**:
  + The code handles cases where an item is not found when attempting to remove or modify it, displaying appropriate messages as required by the assignment.
* **User Interaction**:
  + The print\_menu() function keeps prompting the user until they choose to quit, ensuring continuous interaction with the shopping cart.

Screenshots:

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated