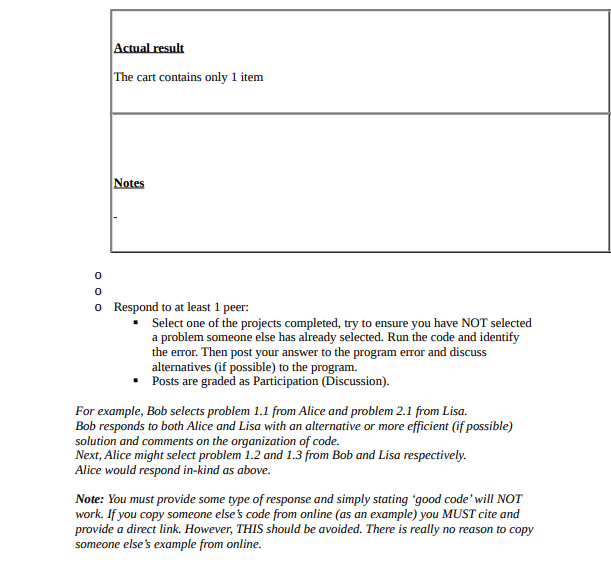


A screenshot of a computer

Description automatically generated A white board with black lines

Description automatically generated A screenshot of a computer

Description automatically generated 

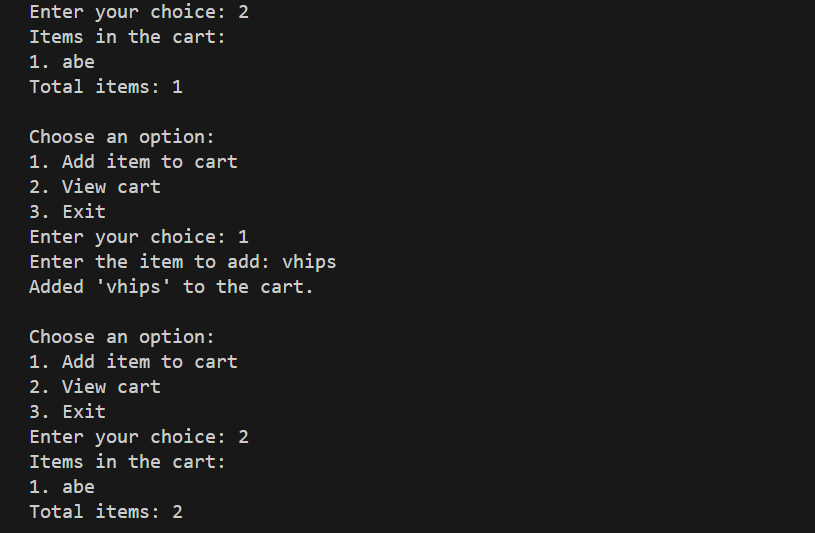
The application's Shopping Cart class incorrectly displayed only the first item in the cart. Debugging revealed the issue in the view cart method, which failed to loop through all items for display. Adjusting the method to iterate through all items resolved the problem. Doctests were added to verify the updated methods.

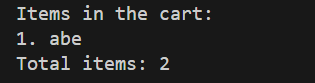
Steps reproduced

**Step 1**

Finding: The application uses a Shopping Cart class that maintains a list of items.

Identified Issue: The view\_cart method displays only the first item instead of all items in the cart.

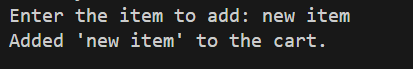




**Step 2**

Finding: The add\_item method adds an item to the self.items list.

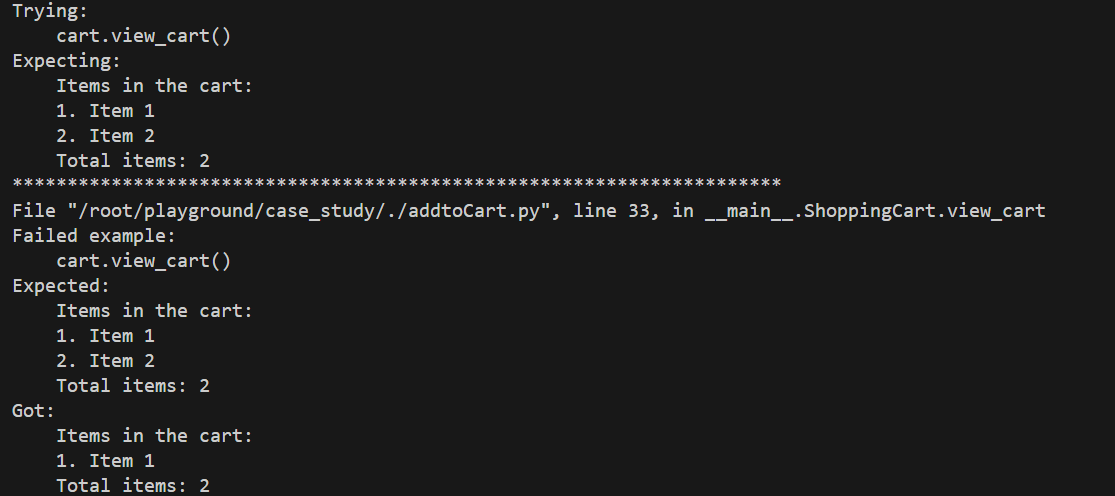
Issue Detection: The issue is not in the add\_item method, as it adds items correctly to the cart.

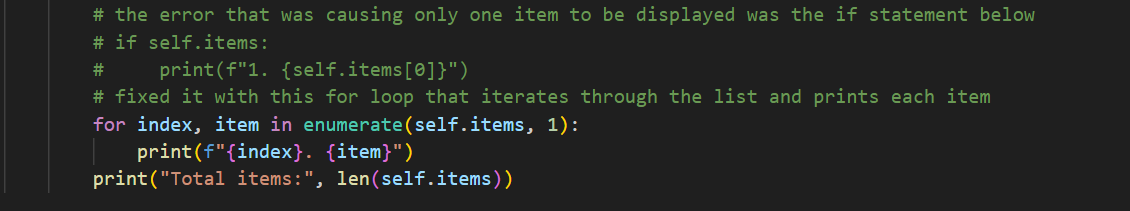


**Step 3**

Finding: The view\_cart method doesn't loop through all items in the cart to display them.

Identified Issue: The method should be modified to loop through all items and display each item in the cart.

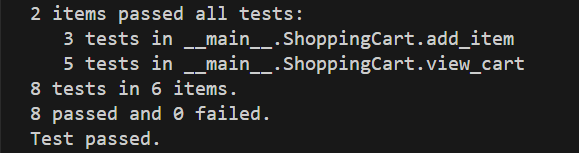


****

**Step 4**

Modification Made: The view\_cart method is revised to loop through all items and display each item in the cart.

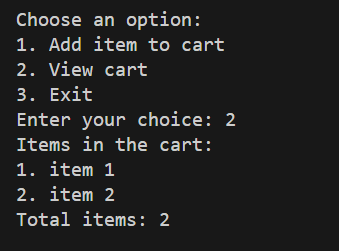
Testing Performed: Doctests are added to validate the add\_item and view\_cart methods in the ShoppingCart class.



**Step 5**

Outcome: After the modification, the view\_cart method successfully displays all items in the cart.

Verification: Doctests are run to confirm that items are correctly added to the cart and the cart displays all items when viewed.



**Implementation**

#!/usr/bin/python3

"""simple program that allows users to add items to a shopping cart and view the items in the cart."""

class ShoppingCart:

    """a shopping cart class"""

    def \_\_init\_\_(self):

        """initialize the cart with an empty list of items"""

        self.items = []

    def add\_item(self, item):

        """

        Add an item to the cart.

        >>> cart = ShoppingCart()

        >>> cart.add\_item('Item 1')

        Added 'Item 1' to the cart.

        >>> cart.add\_item('Item 2')

        Added 'Item 2' to the cart.

        """

        self.items.append(item)

        print(f"Added '{item}' to the cart.")

    def view\_cart(self):

        """

        View the items in the cart.

        >>> cart = ShoppingCart()

        >>> cart.add\_item('Item 1')

        Added 'Item 1' to the cart.

        >>> cart.view\_cart()

        Items in the cart:

        1. Item 1

        Total items: 1

        >>> cart.add\_item('Item 2')

        Added 'Item 2' to the cart.

        >>> cart.view\_cart()

        Items in the cart:

        1. Item 1

        2. Item 2

        Total items: 2

        """

        print("Items in the cart:")

        # the error that was causing only one item to be displayed was the if statement below

        # if self.items:

        #     print(f"1. {self.items[0]}")

        # fixed it with this for loop that iterates through the list and prints each item

        for index, item in enumerate(self.items, 1):

            print(f"{index}. {item}")

        print("Total items:", len(self.items))

def main():

    """main function"""

    cart = ShoppingCart()

    while True:

        print("\nChoose an option:")

        print("1. Add item to cart")

        print("2. View cart")

        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':

            item = input("Enter the item to add: ")

            cart.add\_item(item)

        elif choice == '2':

            cart.view\_cart()

        elif choice == '3':

            print("Exiting the program. Goodbye!")

            break

        else:

            print("Invalid choice. Please enter a valid option.")

if \_\_name\_\_ == "\_\_main\_\_":

    """run the doctests and then the main function"""

    import doctest

    doctest.testmod()

    main()