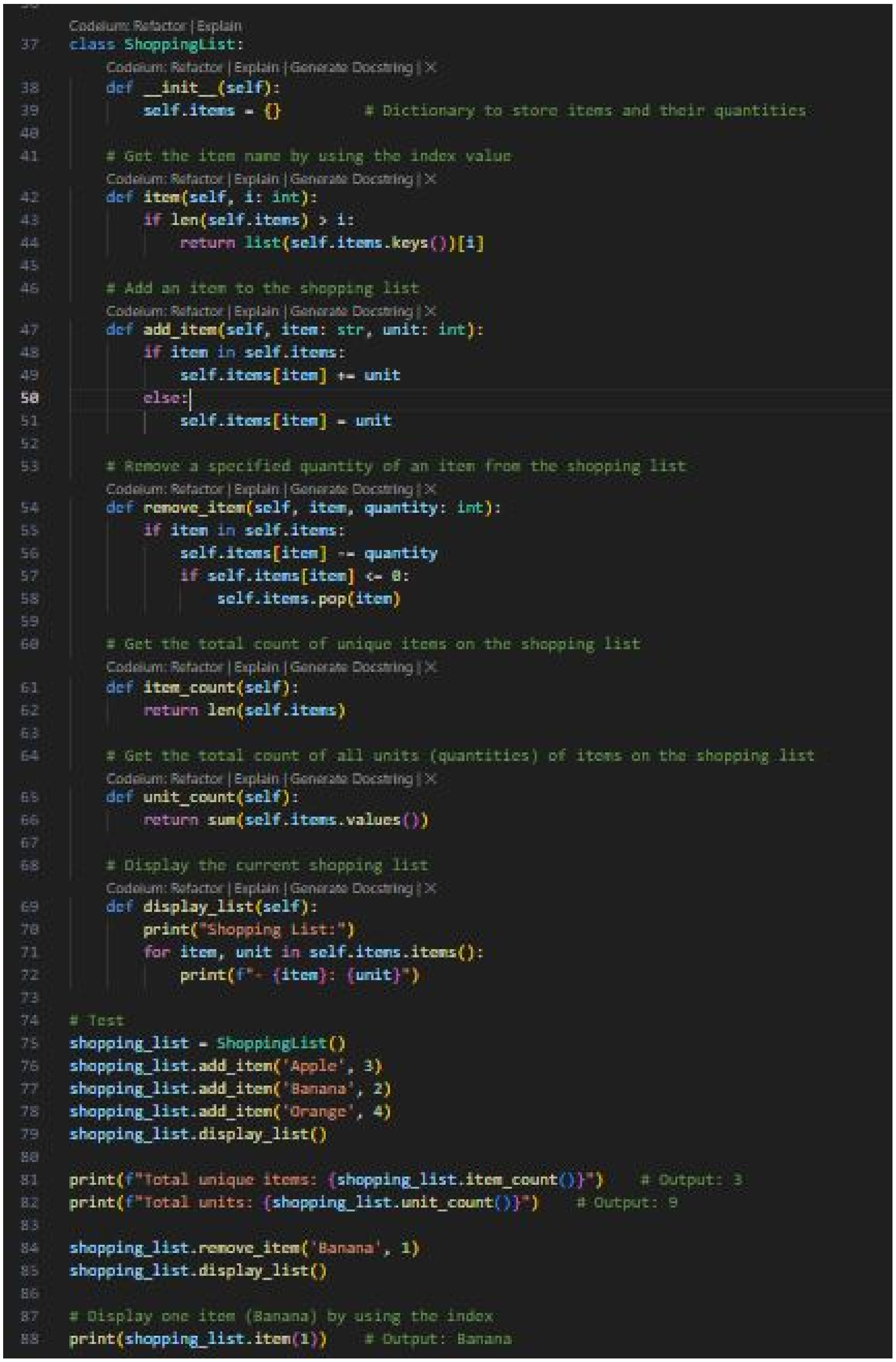
Assignment 4. Object Oriented Programming

Nerea Salamero Labara

March 7, 2025

1 Code

```
# ************************************************
# Title:        Asignments 4
# Author:       Nerea Salamero Labara
# Date:         29/01/2025
# File:         assignment_4.py
# Subject:      Object Oriented Programming
# Description:  This ShoppingList class has methods for adding items, removing items,
#               getting the count of unique items, getting the total units, and
#               displaying the current shopping list etc. Create a ShoppingList class
#               which has several methods item_count, add_item, unit_count etc. Here
#               is partially created shopping list class.
# ************************************************

class ShoppingList:
    def __init__(self):
        self.items = {}         # Dictionary to store items and their quantities

    # Get the item name by using the index value
    def item(self, i: int):
        if len(self.items) > i:
            return list(self.items.keys())[i]

    # Add an item to the shopping list
    def add_item(self, item: str, unit: int):
        if item in self.items:
            self.items[item] += unit
        else:
            self.items[item] = unit

    # Remove a specified quantity of an item from the shopping list
    def remove_item(self, item, quantity: int):
        if item in self.items:
            self.items[item] -= quantity
            if self.items[item] <= 0:
                self.items.pop(item)

    # Get the total count of unique items on the shopping list
    def item_count(self):
        return len(self.items)

    # Get the total count of all units (quantities) of items on the shopping list
    def unit_count(self):
        return sum(self.items.values())
    # Display the current shopping list
    def display_list(self):
        print("Shopping List:")
        for item, unit in self.items.items():
            print(f"- {item}: {unit}")

# Test
shopping_list = ShoppingList()
shopping_list.add_item('Apple', 3)
shopping_list.add_item('Banana', 2)
shopping_list.add_item('Orange', 4)
shopping_list.display_list()

print(f"Total unique items: {shopping_list.item_count()}")    # Output: 3
print(f"Total units: {shopping_list.unit_count()}")    # Output: 9

shopping_list.remove_item('Banana', 1)
shopping_list.display_list()

# Display one item (Banana) by using the index
print(shopping_list.item(1))    # Output: Banana
```

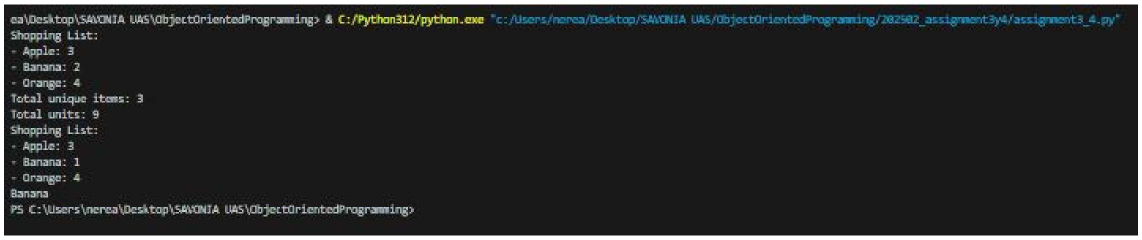

Codeium: Refactor | Explain
Codeium: Refactor | Explain | Generate Docstring | ×

```python
class ShoppingList:
    def __init__(self):
        self.items = {}         # Dictionary to store items and their quantities

    # Get the item name by using the index value
    def item(self, i: int):
        if len(self.items) > i:
            return list(self.items.keys())[i]

    # Add an item to the shopping list
    def add_item(self, item: str, unit: int):
        if item in self.items:
            self.items[item] += unit
        else:
            self.items[item] = unit

    # Remove a specified quantity of an item from the shopping list
    def remove_item(self, item, quantity: int):
        if item in self.items:
            self.items[item] -= quantity
            if self.items[item] <= 0:
                self.items.pop(item)

    # Get the total count of unique items on the shopping list
    def item_count(self):
        return len(self.items)

    # Get the total count of all units (quantities) of items on the shopping list
    def unit_count(self):
        return sum(self.items.values())

    # Display the current shopping list
    def display_list(self):
        print("Shopping List:")
        for item, unit in self.items.items():
            print(f"- {item}: {unit}")

# Test
shopping_list = ShoppingList()
shopping_list.add_item('Apple', 3)
shopping_list.add_item('Banana', 2)
shopping_list.add_item('Orange', 4)
shopping_list.display_list()

print(f"Total unique items: {shopping_list.item_count()}")    # Output: 3
print(f"Total units: {shopping_list.unit_count()}")    # Output: 9

shopping_list.remove_item('Banana', 1)
shopping_list.display_list()

# Display one item (Banana) by using the index
print(shopping_list.item(1))    # Output: Banana
```

Figure 1: Code

2 Output

After executing, the output obtained is the following one:



```
ea\Desktop\SAVONIA UAS\ObjectOrientedProgramming> & C:/Python312/python.exe "c:/Users/nerea/Desktop/SAVONIA UAS/ObjectOrientedProgramming/202502_assignment3y4/assignment3_4.py"
Shopping List:
- Apple: 3
- Banana: 2
- Orange: 4
Total unique items: 3
Total units: 9
Shopping List:
- Apple: 3
- Banana: 1
- Orange: 4
Banana
PS C:\Users\nerea\Desktop\SAVONIA UAS\ObjectOrientedProgramming>
```

Figure 2: Output obtained