Case Study 1- Campus Café Checkout

Step 1: Understanding

A Python program is required to be built that simulates a simple checkout system for the campus café. A customer can view menu, add items to cart, view cart and finally checkout. The program continues till the user wants to exit. The program must show line items, subtotal, tax (10%) and finally the total at the checkout. It can also apply a student's discount (5%) if eligible. For the optional requirements, it can also consider the number of items ordered and process accordingly. It can also display the unique categories in a set and apply a meal deal if both food and drink is present.

Step 2: Inputs & Outputs

The inputs of the program are as follows.

1. Menu Selection (int): Select an option (show menu, add item, view cart, checkout, exit)
2. Item Selection (str): Select the desired item
3. Student Discount (bool): Yes/No
4. For **stretch features**: Takes the number of items as input (int)

The outputs of the program are as follows:

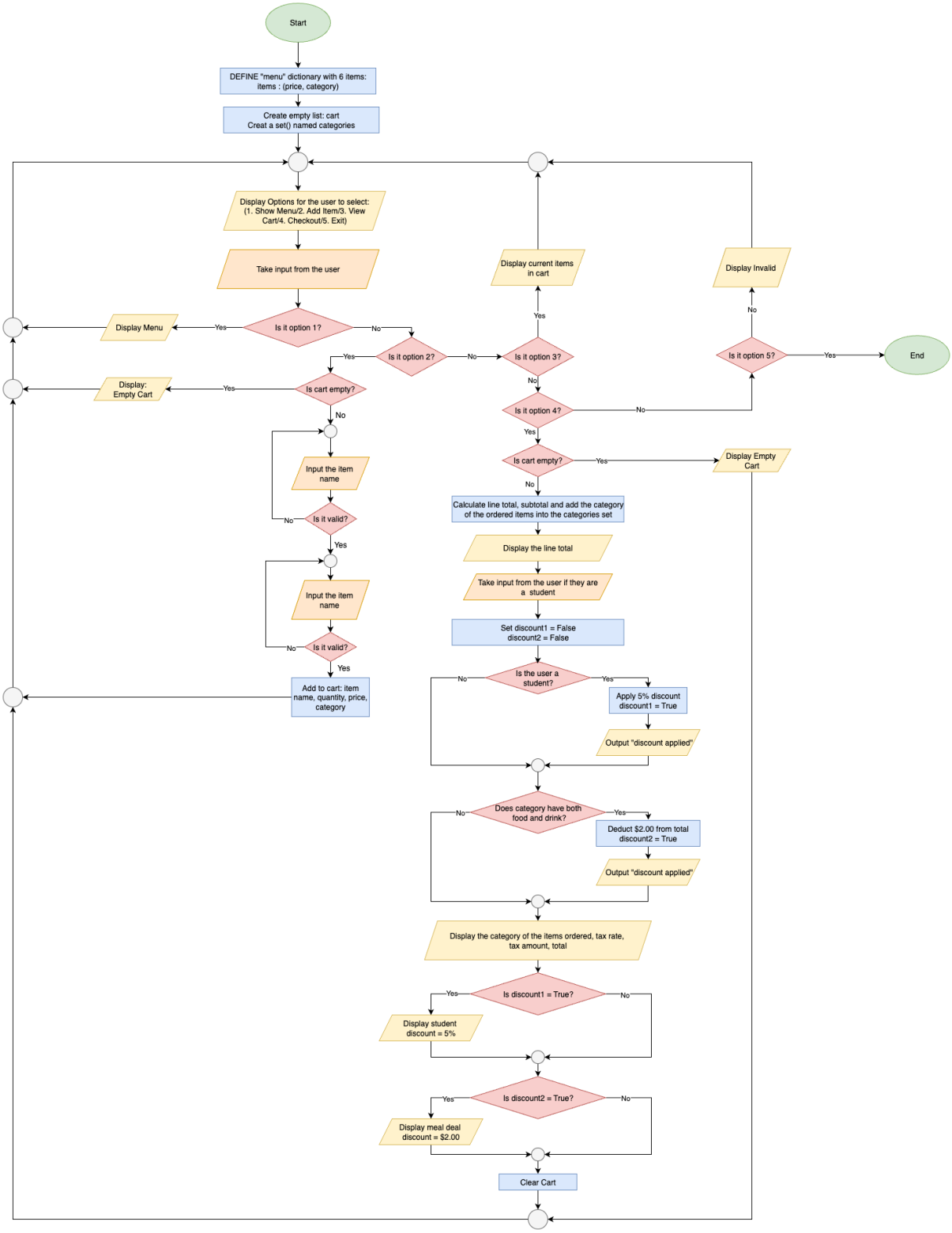
1. Café Menu (str): The available items
2. Cart contents (str): Display the contents currently in the cart whenever asked
3. Receipt (str): Includes line items, subtotal (float, 2 decimal places), tax (10%), tax amount (float, 2 decimal places), discount (if eligible) and total price
4. For **stretch features**: Displays the unique categories, number of items ordered and additional discount for a meal deal

Step 3: Algorithm Design

1. Input:
    a. Menu items from a predefined dictionary that contains all the item names as keys and (price, category) as values
    b. User selection through input()
    c. Input validation
    d. Student Discount (Y/N)
2. Process:
    a. Start with an empty cart[]
    b. While loop: keeps the program going till the user wants to exit and dispays the following functions
        i. “Show Menu”: Displays the Menu of the Café
        ii. “Add item”: Asks for item name and quantity and then appends them to the “cart” list
        iii. “View Cart”: Shows the current cart contents – item name, quantity, price, category
        iv. “Checkout”:
            1. Calculates the line total, subtotal, tax amount, total
            2. Applies student discount (if applicable) and then updates the total (deducts 5%)
            3. If “food” & “drink” in the categories set then applies “Meal-Deal Discount” (deducts $2.00 from total)
            4. Shows which discounts are applicable
            5. Prints the receipt
        v. “Exit”: Breaks the loop and ends the program
    c. Output: Receipt with
        i. Category of items ordered
        ii. Line total
        iii. Subtotal
        iv. Tax rate
        v. Tax Amount
        vi. Student Discount Rate (if applicable)
        vii. Meal-Deal Discount Amount (if applicable)
        viii. Total

Step 4: Pseudocode & Flowchart

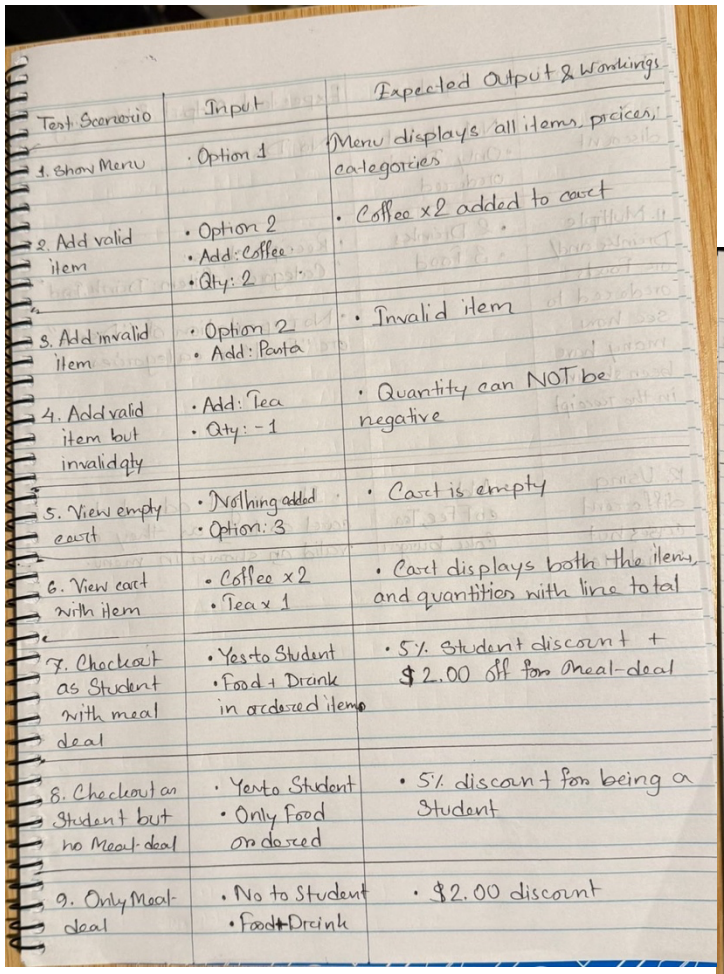
Flowchart:



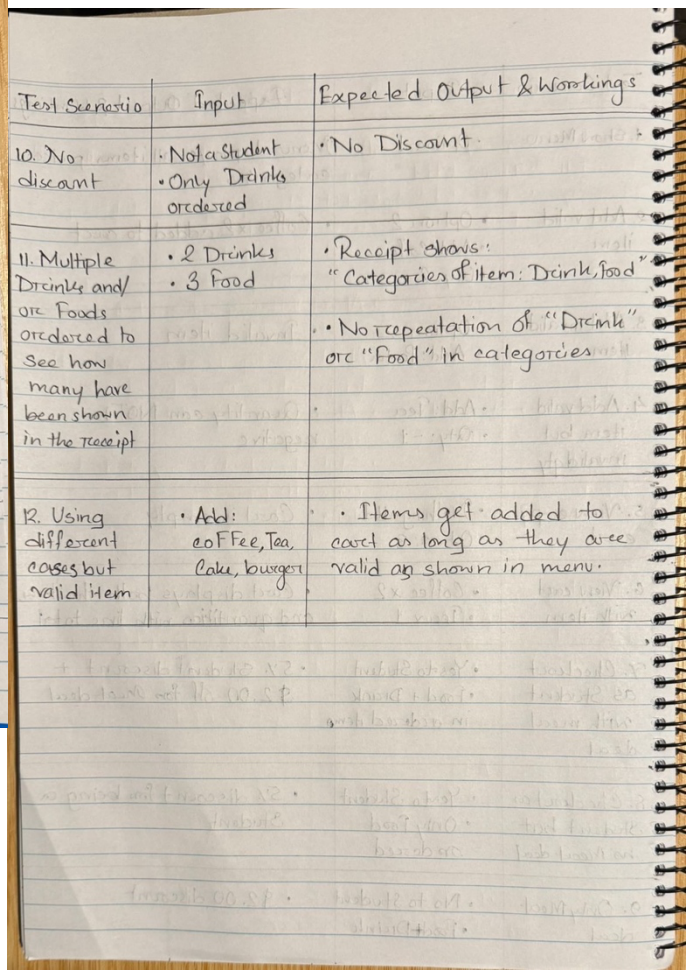
Pseudocode:

```
Begin
    SET menu as dict of “items” : (price, “category”)
    cart <- empty list
    WHILE True
        PRINT "1. Show Menu, 2. Add Item, 3. View Cart, 4. Checkout, 5. Exit"
        GET choice

        IF choice == 1 THEN
        PRINT all items from menu

        ELSE IF choice == 2 THEN
        GET item name
            IF item in menu THEN
                GET quantity
                    IF quanity > 0 THEN
                    ADD (item, price, qty, category) to cart
                    ELSE
                    PRINT "Invalid"
            ELSE
            PRINT "Invalid item"

        ELSE IF choice == 3 THEN
        PRINT all items in cart

        ELSE IF choice == 4 THEN
            subtotal = 0
            categories = empty set
            FOR each (item, price, qty, category) in cart
                Subtotal ← subtotal + (price * qty)
                ADD category to categories
            Tax = 0.10
            Tax_amount ← subtotal * 0.10
            total ← subtotal + tax
            ASK "Student discount? (y/n)"
            IF yes THEN total ← total – (total*0.05)
            IF "Food" in categories AND "Drink" in categories THEN
                total ← total - 2.00
            PRINT receipt with details
            CLEAR cart
        ELSE IF choice == 5 THEN
            BREAK -> END WHILE
END
```

Step 6: Testing- Handwritten Expected Results + Test Runs & Notes

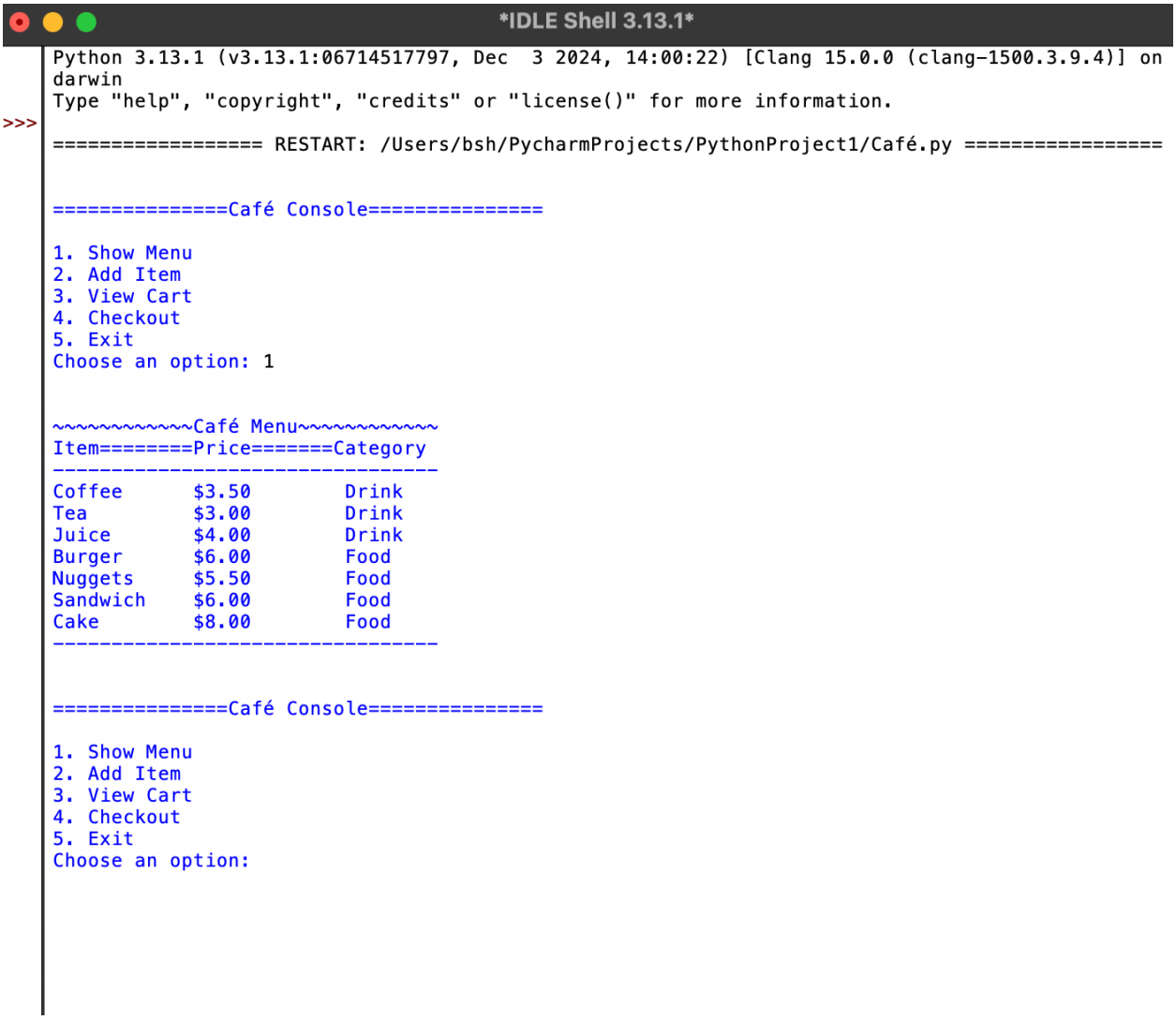


Test Scenario	Input	Expected Output & Workings
1. Show Menu	• Option 1	• Menu displays all items, prices, categories
2. Add valid item	• Option 2 • Add: Coffee • Qty: 2	• Coffee x2 added to cart
3. Add invalid item	• Option 2 • Add: Pasta	• Invalid item
4. Add valid item but invalid qty	• Add: Tea • Qty: -1	• Quantity can NOT be negative
5. View empty cart	• Nothing added • Option: 3	• Cart is empty
6. View cart with item	• Coffee x2 • Tea x 1	• Cart displays both the items, and quantities with line total
7. Checkout as Student with meal deal	• Yes to Student • Food + Drink in ordered items	• 5% Student discount + $2.00 off for meal-deal
8. Checkout as Student but no Meal-deal	• Yes to Student • Only Food ordered	• 5% discount for being a Student
9. Only Meal-deal	• No to Student • Food+Drink	• $2.00 discount



Test Scenario	Input	Expected Output & Workings
10. No discount	• Not a Student • Only Drinks ordered	• No Discount
11. Multiple Drinks and/ or Foods ordered to see how many have been shown in the receipt	• 2 Drinks • 3 Food	• Receipt shows: "Categories of item: Drink, Food" • No repeatation of "Drink" or "Food" in categories
12. Using different cases but valid item	• Add: coFFee, Tea, Cake, burger	• Items get added to cart as long as they are valid as shown in menu.

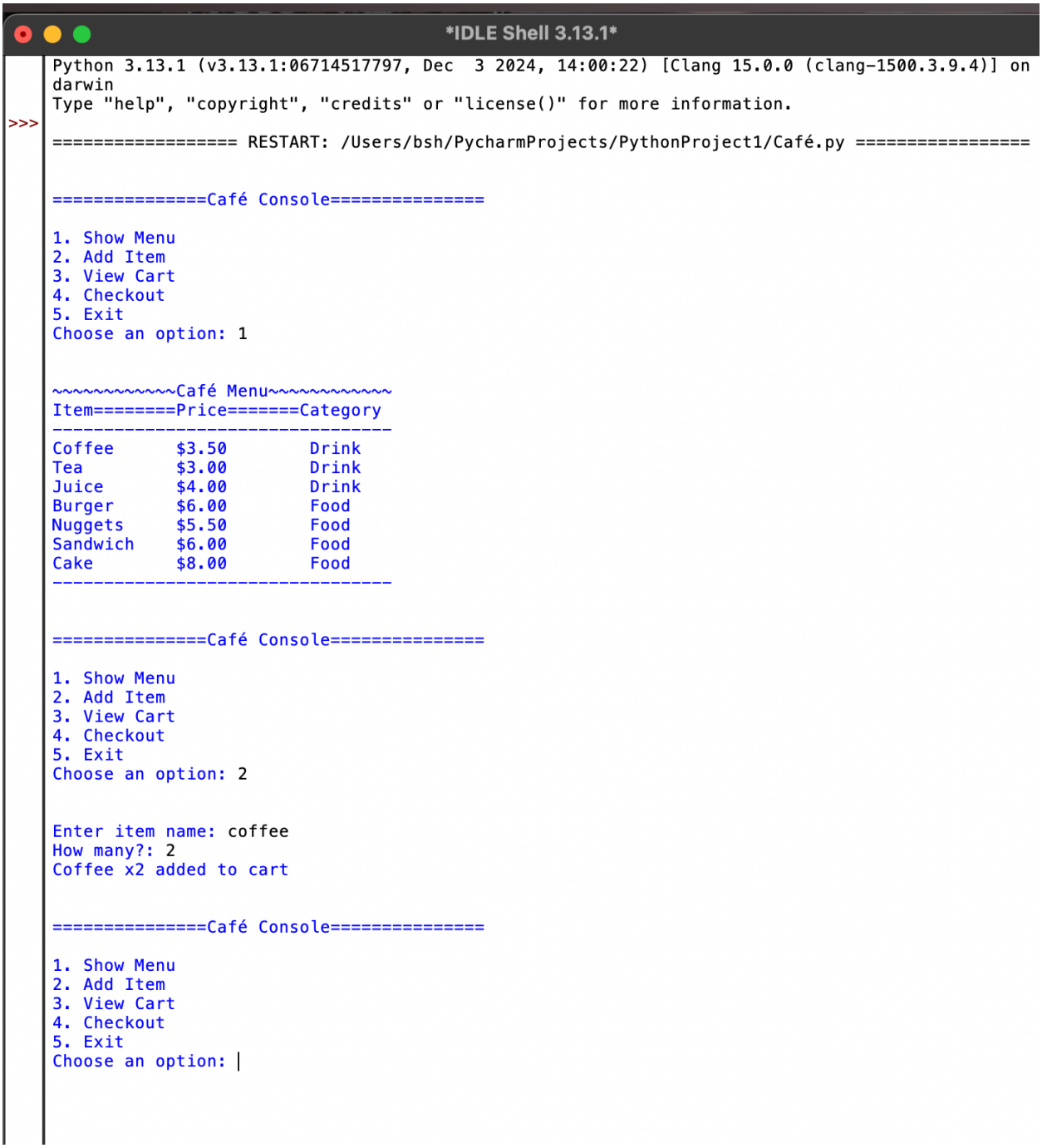
1. Show Menu:
   Note: Worked as Expected
   Test Run:



2. Add Valid Item:

   Note: Worked as Expected

   Test Run:



3. Add Invalid Item:
   Note: Worked as Expected
   Test Run:

```
===============Café Console===============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 2


Enter item name: pasta
Invalid Item
```

4. Add valid item but invalid quantity:
   Note: Worked as Expected
   Test Run:

```
===============Café Console===============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 2


Enter item name: tea
How many?: -2
Quantity can NOT be negative
```

5. View Empty cart:
   Note: Worked as Expected
   Test Run:

```
================Café Console================

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 3
Cart is empty
```

6. View Cart with Item:
   Note: Worked as Expected
   Test Run:

```
===============Café Console===============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 2


Enter item name: tea
How many?: 1
Tea x1 added to cart


===============Café Console===============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 2


Enter item name: burger
How many?: 4
Burger x4 added to cart


===============Café Console===============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 3


==============Your Cart=================
Item--------Qty------Price------Category
========================================
Tea          1       3.00        Drink
Burger       4       24.00       Food
```

7. Checkout as Student with Meal Deal:
   Note: Worked as Expected
   Test Run:

```
======================Receipt==============================
tea         x1         - Price: $3.00      = Line total: 3.00
burger      x4         - Price: $6.00      = Line total: 24.00
Are you a student? Y/N: y


You got a 5% student discount!
You got a $2.00 meal-deal (food + drink) discount!


Categories of items:  Food,Drink
Subtotal :  27.00
Tax : 10%
Tax_amount :  2.70
Student Discount: 5%
Meal-Deal discount: $2.00
Total:  26.21
```

8. Checkout as Student but no Meal Deal:
   Note: Worked as Expected
   Test Run:

```
======================Receipt==============================
tea         x2         - Price: $3.00      = Line total: 6.00
Are you a student? Y/N: y


You got a 5% student discount!


Categories of items:  Drink
Subtotal :  6.00
Tax : 10%
Tax_amount :  0.60
Student Discount: 5%
Total:  6.27
```

9. Only Meal Deal:

   Note: Worked as Expected

   Test Run:

```
========================Receipt==============================
burger      x3          - Price: $6.00      = Line total: 18.00
tea         x2          - Price: $3.00      = Line total: 6.00
Are you a student? Y/N: n
You got a $2.00 meal-deal (food + drink) discount!


Categories of items:  Drink,Food
Subtotal :  24.00
Tax : 10%
Tax_amount :  2.40
Meal-Deal discount: $2.00
Total:  24.40
```

10. No Discount:

    Note: Worked as Expected

    Test Run:

```
========================Receipt==============================
tea         x1          - Price: $3.00      = Line total: 3.00
Are you a student? Y/N: n


Categories of items:  Drink
Subtotal :  3.00
Tax : 10%
Tax_amount :  0.30
Total:  3.30
```

11. Multiple Drinks and/or Foods ordered to see how many have been shown in the receipt:
    Note: Worked as Expected
    Test Run:

```
======================Receipt============================
tea         x2        - Price: $3.00     = Line total: 6.00
coffee      x3        - Price: $3.50     = Line total: 10.50
cake        x2        - Price: $8.00     = Line total: 16.00
burger      x2        - Price: $6.00     = Line total: 12.00
Are you a student? Y/N: y


You got a 5% student discount!
You got a $2.00 meal-deal (food + drink) discount!


Categories of items:  Drink,Food
Subtotal :  44.50
Tax : 10%
Tax_amount :  4.45
Student Discount: 5%
Meal-Deal discount: $2.00
Total:  44.50
```

12. Using different cases but valid item:
    Note: Worked as Expected
    Test Run:

```
==============Café Console==============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 2


Enter item name: coFFee
How many?: 2
Coffee x2 added to cart


==============Café Console==============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 2


Enter item name: Tea
How many?: 2
Tea x2 added to cart


==============Café Console==============

1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Choose an option: 2


Enter item name: burger
How many?: 3
Burger x3 added to cart
```

Step 7: Refinement via GenAI

I have used Microsoft Copilot to refine my code and the prompt that I used to do so:

“This is my Python café checkout program. Can you give me suggestions to make it cleaner and easier to use? Especially input validation like stopping wrong quantities) and making the receipt and the display menu/tables look more professional. Keep the main structure but just make it good looking. Don’t remove the student discount or meal deal features.”

Before (issues in my code):

- Input validation was weak (quantities not checked properly)
- Receipt formatting looked unaligned
- Cart didn’t get cleared after the user was done with checkout

After (GenAI suggested changes):

- Strengthened validation: quantity must be an integer > 0, otherwise the program re-prompts
- Improved receipt formatting with alignment print("Item" + "-"*8 + "Qty" + "-"*6 + "Price" + "-"*6 + "Category")
- Added error messages (e.g., “Quantity can NOT be negative”)
- Added cart.clear() at the end of the Checkout function

Justification (why I accepted changes):

- Stronger input validation prevents crashes and improves user experience
- Cleaner receipt layout is more professional
- Clearing cart after checkout ensures no leftovers when starting a new order

I did not accept one suggestion: GenAI proposed adding file saving for receipts. I rejected it because it goes beyond the case study requirements and might complicate marking.