

Bazla Bilquees

U3312671

Introduction To Information Technology

## **ASSIGNMENT # 02 – PYTHON FUNDAMENTAL**

### **CASE STUDY # 01 CAMPUS CAFÉ CHECKOUT**

#### Step 1 — Understand the Problem:

Develop a console point-of-sale application that allows an individual to navigate a menu at a cafe, add items to a collections cart, view the items in the cart and check out. Line items, subtotal, tax up to 10% and optional student discount 5% and a final total must be displayed on the receipt. Use a dictionary to use the menu, a list to use the cart and a set to use categories.

#### Step 2 — Inputs & Outputs

SYMBOL	Name	Type	Role	Description / logic
F	Food present	Boolean	Input	1 if cart contains at least one <b>food</b> item, else 0
D	Drink present	Boolean	Input	1 if cart contains at least one <b>drink</b> item, else 0
S	Student flag	Boolean	Input	1 if user confirms they are a student (student discount applies), else 0
P	Cart non-empty	Boolean	Input	1 if cart has at least 1 item, else 0
M	Meal-deal	Boolean	Output	1 if meal-deal discount applies (food AND drink present)
SD	Student Discount	Boolean	Output	1 if student discount applies (S)
CD	Combined Discount Flag	Boolean	Output	1 if <i>any</i> discount is available (meal-deal OR student)
CO	Checkout allowed	Boolean	Output	1 if checkout can proceed (cart non-empty = P)
PR	Print Receipt	Boolean	Output	1 if CO = 1 (receipt should be printed after checkout)

## Step 3 — Algorithm

1. Start
2. Enter menu items, prices and categories (Drink or Food).
3. Initialize an empty cart.
4. Continue till one decides to quit:
  - 4.1 Display main menu:
    1. Show Menu
    2. Add Item
    3. View Cart
    4. Checkout
    5. Exit
  - 4.2 Read user choice
  - 4.3 If choice = Show Menu:
    - List all menu items including price and category.
  - 4.4 If choice = Add Item:
    - Ask user to enter item name
    - Check if item exists in menu
    - Yes, request quantity (default= 1)
    - Add (item, quantity) to cart
    - Display confirmation
    - If no, display "Item not on menu"
  - 4.5 If choice = View Cart:
    - In case there is nothing in the cart, list "Cart is empty"
    - otherwise, list everything in cart, including quantity and total price.
  - 4.6 If choice = Checkout:
    - In case of cart empty, it will show Nothing to checkout and move on.

- Subtotal = (price x amount) of all items in cart.

- Calculate tax = subtotal × 10%

4.7 If choice = Exit:

- Display "Exiting program"

- Stop loop

4.8 Else:

- Display "Invalid choice"

End program

## Step 4 — Flow Chart



u3312671-bazla  
Bilquees- Assignment

## Step 4 – PSEUDOCODE

BEGIN

define menu items including price and category.

MENU = {

"Coffee": (3.50, "Drink"),

"Tea": (2.50, "Drink"),

"Muffin": (2.00, "Food"),

"Sandwich": (5.00, "Food"),

"Smoothie": (4.00, "Drink"),

"Salad": (4.50, "Food")

}

CART = []

FUNCTION show\_menu()

PRINT "--- Café Menu ---"

FOR each ITEM in MENU

Name, price, category of PRINT ITEM.

END FOR

END FUNCTION

FUNCTION add\_item(CART)

CALL show\_menu()

INPUT item\_name

IF itemname in MENU THEN

INPUT quantity (default 1)

ADD (name of item, amount) to CART.

Print x number of item-name added to cart.

ELSE

PRINT "Item not on the menu"

```

END IF

END FUNCTION

FUNCTION view_cart(CART)
IF CART is empty THEN
PRINT "Cart is empty"
RETURN
END IF

PRINT "--- Current Cart ---"

Qty in CART FOR each item.
price = MENU[item].price
PRINT quantity x item - price*quantity
END FOR

END FUNCTION

FUNCTION checkout(CART)
IF CART is empty THEN
PRINT "Cart is empty. Nothing to checkout."
RETURN
END IF

subtotal = The total of (price x quantity) of items in CART.
tax = subtotal * 0.10

// Student discount
GRAB apply student discount (y/n).
IF yes THEN
discount = (subtotal + tax) * 0.05
ELSE
discount = 0

```

END IF

Promotion: 2 drink discount and food combo: 2 drinks off and food combo.

drinks count = SUM of the quantity of items in the CART whose category is a drink.

foods count = SUM of quantities of items that are in CART and category = Food.

combos = minimum of drinks, foods count

meal\_deal\_discount = combos \* 2.00

subtotal = total, less tax, less discount, less meal deal discount.

// Print receipt

PRINT "--- Receipt ---"

Qty in CART FOR each item.

PRINT quantity x item - price\*quantity

END FOR

PRINT subtotal, tax

IF discount > 0 then PRINT student discount -discount.

IF meal\_deal\_discount >0 then PRINT meal\_deal\_discount -meal\_deal\_discount.

PRINT total

PRINT "Thank you for visiting!"

END FUNCTION

FUNCTION main()

WHILE True

PRINT menu choices: Show Menu, Add Item, View Cart, Checkout, Exit

INPUT user\_choice

SWITCH user\_choice

CASE 1: CALL show\_menu()

CASE 2: CALL add\_item(CART)

CASE 3: CALL view\_cart(CART)

CASE 4: CALL out(CART), and CLEAR CART.

CASE 5: PRINT "Leaving program" BREAK

DEFAULT: PRINT "Illegal option"

END SWITCH

END WHILE

END FUNCTION

ALL main()

END

## Step 5 Truth Table

F	D	K	S	T	P	N	MULTICAT	MEAL	DISC	RECEIPT	CHECKOUT_OK
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	0	0	1	1	0	0	0	1	1
0	0	1	0	1	0	1	0	0	0	1	0
0	0	1	0	1	1	1	0	0	0	1	1
0	0	1	1	0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0	0	1	1	1
0	0	1	1	1	0	1	0	0	1	1	0
0	0	1	1	1	1	1	0	0	1	1	1
0	1	0	0	0	0	1	0	0	0	1	0
0	1	0	0	0	1	1	0	0	0	1	1
0	1	0	0	1	0	1	0	0	0	1	0
0	1	0	0	1	1	1	0	0	0	1	1
0	1	0	1	0	0	1	0	0	1	1	0
0	1	0	1	1	1	1	0	0	1	1	1
0	1	1	0	0	0	1	1	0	0	1	0
0	1	1	0	0	1	1	1	0	0	1	1

0	1	1	0	1	0	1	1	0	0	1	0
0	1	1	0	1	1	1	1	0	0	1	1
0	1	1	1	0	0	1	1	0	1	1	0
0	1	1	1	0	1	1	1	0	1	1	1
0	1	1	1	1	0	1	1	0	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1
1	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	1	1	0	0	0	1	1
1	0	0	0	1	0	1	0	0	0	1	0
1	0	0	0	1	1	1	0	0	0	1	1
1	0	0	1	0	0	1	0	0	1	1	0
1	0	0	1	0	1	1	0	0	1	1	1
1	0	0	1	1	0	1	0	0	1	1	0
1	0	0	1	1	1	1	0	0	1	1	1
1	0	1	0	0	0	1	1	0	0	1	0
1	0	1	0	0	1	1	1	0	0	1	1
1	0	1	0	1	0	1	1	0	0	1	0
1	0	1	0	1	1	1	1	0	0	1	1
1	0	1	1	0	0	1	1	0	1	1	0
1	0	1	1	0	1	1	1	0	1	1	1
1	0	1	1	1	0	1	1	0	1	1	0
1	0	1	1	1	1	1	1	0	1	1	1
1	1	0	0	0	0	1	1	0	0	1	0
1	1	0	0	0	1	1	1	0	0	1	1
1	1	0	0	1	0	1	1	1	0	1	0
1	1	0	0	1	1	1	1	1			

## Step 6 Logic Diagram



u3312671-Bazla  
Bilquees-Assignment

## Step 7 Python Code

```
# Name: Bazla Bilquees
# Student ID: u3312671
# Case Study: Campus Café Checkout

# Prices menu: item -> (price, category)
menu = {
    "Coffee": (3.50, "Drink"),
    "Tea": (2.50, "Drink"),
```



```

    "Muffin": (2.00, "Food"),
    "Sandwich": (5.00, "Food"),
    "Smoothie": (4.00, "Drink"),
    "Salad": (4.50, "Food")
}

cart = []

# Function to display the menu
def show_menu():
    print("\n--- Café Menu ---")
    for item, (price, category) in menu.items():
        print(f'{item:10} - ${price:.2f} ({category})')
    print("-----")

# Add item to cart
def add_item(cart):
    show_menu()
    item = input("Enter item to add: ").title()
    if item in menu:
        qty = input("Quantity (default 1): ").strip()
        qty = int(qty) if qty.isdigit() and int(qty) > 0 else 1
        cart.append((item, qty))
        print(f'{qty} x {item} added to cart.')
    else:
        print("Item not on the menu.")

# View current cart
def view_cart(cart):
    if not cart:
        print("Cart is empty.")
        return
    print("\n--- Current Cart ---")
    for item, qty in cart:
        price = menu[item][0]
        print(f'{qty} x {item} - ${price*qty:.2f}')
    print("-----")

# Checkout and print receipt
def checkout(cart):
    if not cart:
        print("Cart is empty. Nothing to checkout.")
        return

    subtotal = sum(menu[item][0]*qty for item, qty in cart)
    tax = subtotal * 0.10

```

```

discount = 0.0

# Student discount
student = input("Apply student discount (5%)? (y/n): ").lower()
if student == "y":
    discount = (subtotal + tax) * 0.05

# Meal deal discount: $2 off per drink+food combo
drinks = sum(qty for item, qty in cart if menu[item][1] == "Drink")
foods = sum(qty for item, qty in cart if menu[item][1] == "Food")
meal_deal_discount = 0
combos = min(drinks, foods)
if combos >= 1:
    meal_deal_discount = combos * 2.00

total = subtotal + tax - discount - meal_deal_discount

# Print receipt
print("\n--- Receipt ---")
for item, qty in cart:
    price = menu[item][0]
    print(f"{qty} x {item} - ${price*qty:.2f}")
print(f"Subtotal: ${subtotal:.2f}")
print(f"Tax (10%): ${tax:.2f}")
if discount > 0:
    print(f"Student discount: -${discount:.2f}")
if meal_deal_discount > 0:
    print(f"Meal deal discount: -${meal_deal_discount:.2f}")
print(f"Total: ${total:.2f}")
print("Thank you for visiting!")

# Main program loop
def main():
    while True:
        print("\n--- Café POS ---")
        print("1. Show Menu\n2. Add Item\n3. View Cart\n4. Checkout\n5. Exit")
        choice = input("Enter choice: ").strip()
        if choice == "1":
            show_menu()
        elif choice == "2":
            add_item(cart)
        elif choice == "3":
            view_cart(cart)
        elif choice == "4":
            checkout(cart)
        cart.clear()

```

```
elif choice == "5":  
    print("Exiting program.")  
    break  
else:  
    print("Invalid choice. Try again.")  
  
if __name__ == "__main__":  
    main()
```

Step 8 -Testing: handwritten expected results + test runs & notes

## Case Study -1 Assignment 2.

Run:

STEP #06.

1. Show menu:
2. Add Item
3. View Cart
4. Checkout
5. Exit.

After choosing option 1 it will show the menu with price.

Coffee — \$3.50  
 Tea — \$2.50  
 Muffin — \$2.00  
 Sandwich — \$5.00  
 Smoothie — \$4.00  
 Salad — \$4.50

After that when we choose option 2 to add item into the cart it will ask the quantity of that item for example if i choose tea it will show me like this:

Enter item to add: Tea  
 Quantity = 4

If i want to add another item i have to repeat the above procedure after that when i choose option 3 it will show the display:

4 x Tea = \$10.00

4 x Muffin = \$8.00

After choosing option 4 it will ask about the student discount (5%)

4 x Tea = \$10.00

4 x Muffin = \$8.00

Subtotal = \$18.00

Tax (10%) = \$1.80

Student discount (5%) = \$-0.99

Meal-deal discount = \$8.00

Total = \$10.81

∴ Meal deal discount  
 when food and drink  
 both present into the  
 cart.

```
IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
Python 3.13.7 (tags/v3.13.7:bce1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>> = RESTART: C:/Users/Lenovo/Desktop/u3312671-Bazla Bilquees- Assignment 2- Case study 1.py

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 1

--- Café Menu ---
Coffee - $3.50 (Drink)
Tea - $2.50 (Drink)
Muffin - $2.00 (Food)
Sandwich - $5.00 (Food)
Smoothie - $4.00 (Drink)
Salad - $4.50 (Food)

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 2

--- Café Menu ---
Coffee - $3.50 (Drink)
Tea - $2.50 (Drink)
Muffin - $2.00 (Food)
Sandwich - $5.00 (Food)
Smoothie - $4.00 (Drink)
Salad - $4.50 (Food)

Enter item to add: Tea
Quantity (default 1): 4
4 x Tea added to cart.

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 2

--- Café Menu ---
Coffee - $3.50 (Drink)
Tea - $2.50 (Drink)
Muffin - $2.00 (Food)
Sandwich - $5.00 (Food)
Smoothie - $4.00 (Drink)
Salad - $4.50 (Food)

Enter item to add: Muffin
Quantity (default 1): 4
4 x Muffin added to cart.

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 3

--- Current Cart ---
4 x Tea - $10.00
4 x Muffin - $8.00

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 4

--- Current Cart ---
4 x Tea - $10.00
4 x Muffin - $8.00
Total - $18.00

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 5
```

```
IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 3

--- Current Cart ---
4 x Tea - $10.00
4 x Muffin - $8.00
-----

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 4
Apply student discount (5%)? (y/n): y

--- Receipt ---
4 x Tea - $10.00
4 x Muffin - $8.00
Subtotal: $18.00
Tax (10%): $1.80
Student discount: -$0.99
Meal deal discount: -$8.00
Total: $10.81
Thank you for visiting!

--- Café POS ---
1. Show Menu
2. Add Item
3. View Cart
4. Checkout
5. Exit
Enter choice: 5
Exiting program.
>>>
```

Activate Windows  
Go to Settings to activate Windows.

Ln: 103 Col: 0  
5:52 PM  
9/13/2025

## Step 9- Refinement via GenAI: prompt(s), what changed, justification

The Python Café checkout system has been improved with a number of practical features that greatly improve user experience and logic of operation when compared with the initial system. The system used to be limited to calculating a simple subtotal and providing a fixed tax rate as well, which did not offer flexibility and incentives to the customers. The new version has introduced a meal-deal discount and automatically a 10 percent discount is offered in case both food and beverage products are in the cart. This will stimulate the selling of multiple products and capture real world promotion. The other significant upgrade is the optional student discount, whereby users can make sure that they are entitled to an extra 5 percent off. This substitutes the inflexible nature of the old system, which used a discounts calculation without the user involvement. In addition, any money values are now in a uniform format of two decimal places as:.2f, which makes receipt readability and professionalism more agreeable. There is also the introduction of a loyalty points program in the new version where one point is earned after every five dollars are spent. This enhances the check out process and retention of customers. Finally, users have the option of saving their receipt which mimics the real world digital receipt storage. Compared to the original, the new system is more dynamic, easier to use and commercially feasible- it turns a mere calculator into a smart, customer-oriented checkout helper.

Sign in

Python Café Checkout System – Enhanced Version

Overview

This document outlines the improvements made to a basic Python café checkout system. The original version included simple subtotal and tax calculations. The enhanced version introduces:

- Meal-deal discount logic
- Optional student discount
- Consistent currency formatting
- Safer input validation
- Loyalty points system

Message Copilot

Smart (GPT-5)

+

Activate Windows  
Go to Settings to activate Windows.

Search

copilot.microsoft.com/chats/JghnA8q1LPpCs2k8JryqJ

7:31 AM  
9/11/2025

Sign in

- Meal-deal discount logic
- Optional student discount
- Consistent currency formatting
- Safer input validation
- Loyalty points system
- Receipt saving option

Before vs After Comparison

Feature	Before	After
---------	--------	-------

Message Copilot

Smart (GPT-5)

+

Activate Windows  
Go to Settings to activate Windows.

Search

7:32 AM  
9/11/2025

Sign in

Feature	Before	After
Currency Formatting	Inconsistent	All monetary values formatted to two decimals using <code>\$.2f</code>
Meal-Deal Discount	Not present	10% off if cart contains both food and drink
Student Discount	Always prompted	Now optional and applied only if user confirms
Input Validation	Basic	Improved: safer quantity input and choice handling
Loyalty Points	Not available	Earn 1 point per \$5 spent
Receipt Saving	Not available	Option to save receipt after checkout

Message Copilot

Smart (GPT-5)

+

Activate Windows  
Go to Settings to activate Windows.

Sign in

Loyalty Points	Not available	Earn 1 point per \$5 spent
Receipt Saving	Not available	Option to save receipt after checkout

Code Structure

1. Menu Definition

Python

Copy

```
menu = {  
    "Coffee": (3.50, "Drink"),  
    "Tea": (2.75, "Drink"),  
    "Sandwich": (5.00, "Food"),  
    "Salad": (4.50, "Food"),  
    "Cookie": (1.25, "Food"),  
    "Juice": (3.00, "Drink")  
}
```

Message Copilot

Smart (GPT-5)

+

Activate Windows  
Go to Settings to activate Windows.

Sign in

1. Menu Definition

Python

Copy

```
menu = {  
    "Coffee": (3.50, "Drink"),  
    "Tea": (2.75, "Drink"),  
    "Sandwich": (5.00, "Food"),  
    "Salad": (4.50, "Food"),  
    "Cookie": (1.25, "Food"),  
    "Juice": (3.00, "Drink")  
}
```



← → ↻ copilot.microsoft.com/chats/JghnA8q1LPPcs2k8JryqJ ☆ 📌 ⬇️ ⓘ

📄 📄 ⌵

Sign in ⓘ

```
}


```

## 2. Meal-Deal Detection

Python ⌵ Copy

```
def has_meal_deal(cart):
    categories = set(menu[item][1] for item, qty in cart)
    return "Food" in categories and "Drink" in categories


```

## 3. Cart Builder

Python ⌵ Copy

```
def build_cart():


```

Message Copilot

Smart (GPT-5) ⌵ + 🗣️ Activate Windows  
Go to Settings to activate Windows.

🪟 🔍 Search

← → ↻ copilot.microsoft.com/chats/JghnA8q1LPPcs2k8JryqJ ☆ 📌 ⬇️ ⓘ

📄 📄 ⌵

Sign in ⓘ

```
return "Food" in categories and "Drink" in categories


```

## 3. Cart Builder

Python ⌵ Copy

```
def build_cart():
    cart = []
    print("Welcome to the Café! Here's our menu:")
    for item, (price, _) in menu.items():
        print(f"-- {item}: ${price:.2f}")

    while True:
        choice = input("\nEnter item name to add (or 'done' to finish): ").l
        if choice == "Done":
            break
        if choice not in menu:


```

Message Copilot

Smart (GPT-5) ⌵ + 🗣️ Activate Windows  
Go to Settings to activate Windows.

🪟 🔍 Search

← → ↻ copilot.microsoft.com/chats/JghnA8q1LPPcs2k8JryqJ ☆ 📌 ⬇️ ⓘ

📄 📄 ⌵

Sign in ⓘ

```
choice = input("\nEnter item name to add (or 'done' to finish): ").l
if choice == "Done":
    break
if choice not in menu:
    print("Item not found. Please choose from the menu.")
    continue
try:
    qty = int(input(f"Enter quantity for {choice}: "))
    if qty <= 0:
        raise ValueError
except ValueError:
    print("Invalid quantity. Defaulting to 1.")
    qty = 1
cart.append((choice, qty))
return cart


```

← → ↻ copilot.microsoft.com/chats/JghnA8q1LPPcs2k8JryqJ ☆ 🔍 ⬇️ 6 ⋮

📄 📝 ⌵

Sign in 👤

#### 4. Loyalty Points Calculation

```
Python ^ Copy
def calculate_loyalty_points(total):
    return int(total // 5)
```

#### 5. Receipt Generator

```
Python ^ Copy
def generate_receipt(cart, subtotal, tax, discount, total):
    lines = ["--- 📄 Receipt ---"]
    for item, qty in cart:
```

Message Copilot

Smart (GPT-5) ⌵

+ 🔊 Activate Windows  
Go to Settings to activate Windows.

Windows Search 📁 📄 📝 ⌵

← → ↻ copilot.microsoft.com/chats/JghnA8q1LPPcs2k8JryqJ ☆ 🔍 ⬇️ 6 ⋮

📄 📝 ⌵

Sign in 👤

#### 5. Receipt Generator

```
Python ^ Copy
def generate_receipt(cart, subtotal, tax, discount, total):
    lines = ["--- 📄 Receipt ---"]
    for item, qty in cart:
        price = menu[item][0]
        lines.append(f"{qty} x {item} - ${price * qty:.2f}")
    lines.append(f"Subtotal: ${subtotal:.2f}")
    lines.append(f"Tax (10%): ${tax:.2f}")
    lines.append(f"Total Discounts: -${discount:.2f}")
    lines.append(f"Total: ${total:.2f}")
    return "\n".join(lines)
```

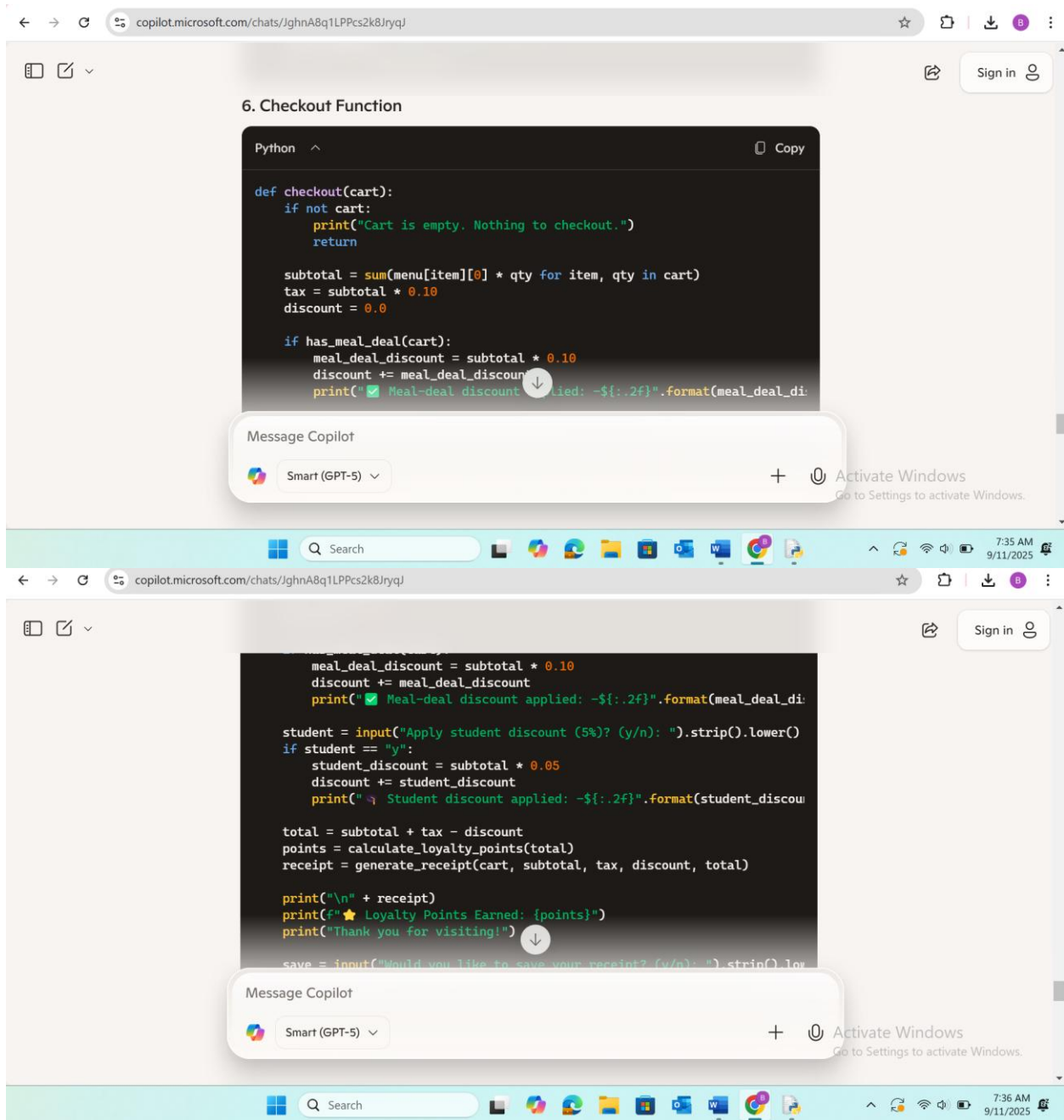
#### 6. Checkout Function

Message Copilot

Smart (GPT-5) ⌵

+ 🔊 Activate Windows  
Go to Settings to activate Windows.

Windows Search 📁 📄 📝 ⌵



← → ↻ copilot.microsoft.com/chats/JghnA8q1LPPcs2k8JryqJ ☆ 🔍 ⬇️ B ⋮

📄 ✍️ ▾ Sign in 👤

```
print(f"★ Loyalty Points Earned: {points}")
print("Thank you for visiting!")

save = input("Would you like to save your receipt? (y/n): ").strip().lower()
if save == "y":
    print("\n📄 Receipt saved:\n")
    print(receipt)
```

### 7. Main Program

Python ^ Copy

```
if __name__ == "__main__":
    cart = build_cart()
    checkout(cart)
```

↓

Message Copilot

Smart (GPT-5) ▾ + 🔊 Activate Windows  
Go to Settings to activate Windows.

← → ↻ copilot.microsoft.com/chats/JghnA8q1LPPcs2k8JryqJ ☆ 🔍 ⬇️ B ⋮

📄 ✍️ ▾ Sign in 👤

### ✅ Sample Interaction

Code ^ Copy

```
Enter item name to add (or 'done' to finish): Coffee
Enter quantity for Coffee: 2
Enter item name to add (or 'done' to finish): Sandwich
Enter quantity for Sandwich: 1
Enter item name to add (or 'done' to finish): done
Apply student discount (5%)? (y/n): y
Would you like to save your receipt? (y/n): y
```

Output:

↓

Code ^ Copy

Message Copilot

Smart (GPT-5) ▾ + 🔊 Activate Windows  
Go to Settings to activate Windows.

