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 **food** item, else 0 |
| D | Drink present | Boolean | Input | 1 if cart contains at least one **drink** 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 *any* 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

## Start

## Enter menu items, prices and categories (Drink or Food).

## 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



## 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 "Illlegal 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 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 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

## 

## 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

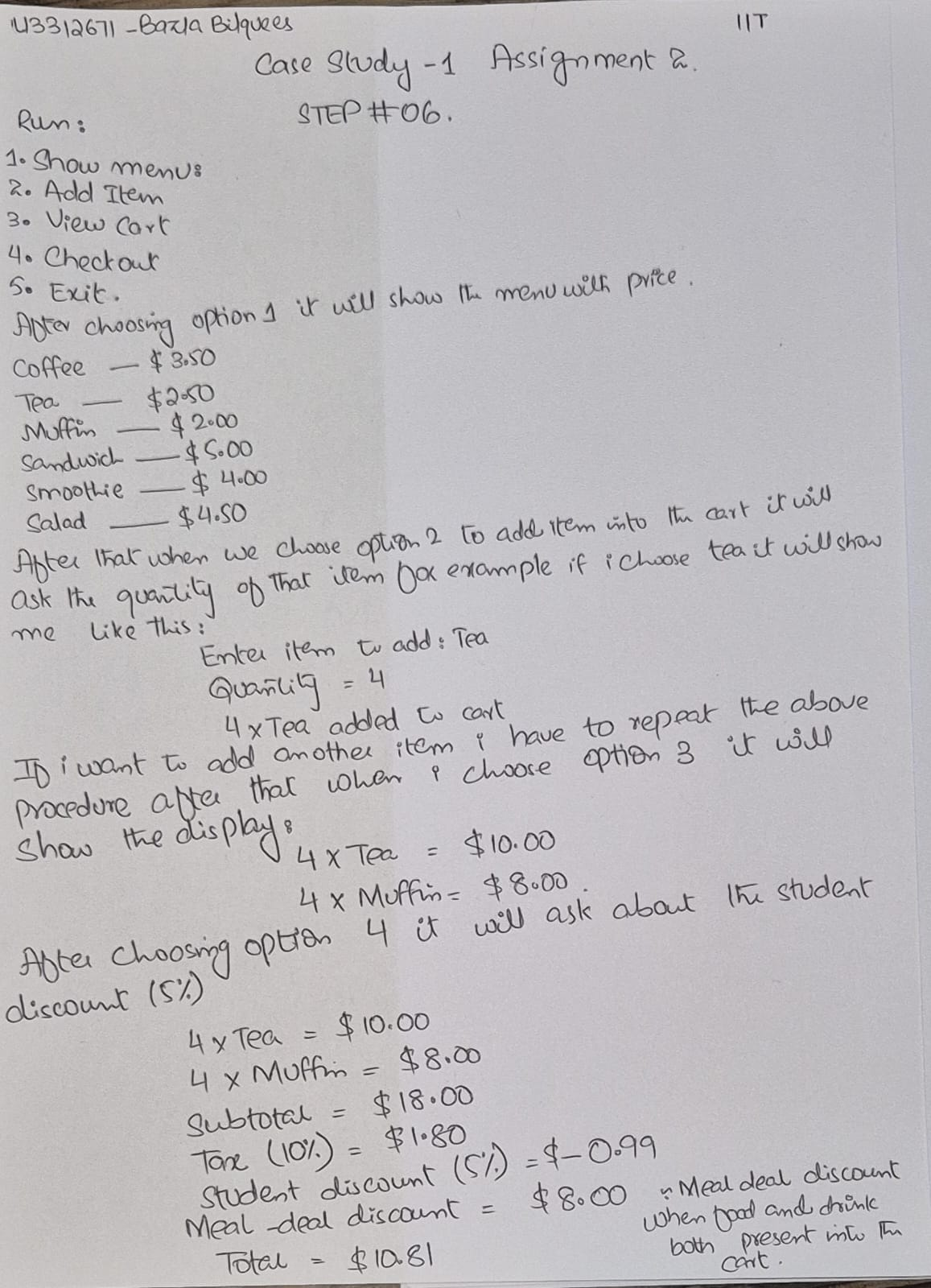
else:

print("Invalid choice. Try again.")

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

main()

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



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

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

The Python Cafe 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.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.