Portfolio Project Pt I: Shopping Cart Class

# Source Code

Below is the Python source code that implements the ItemToPurchase Class and main function.

## Part 1: ItemToPurchase Class

'''

Program Outline: Shopping Cart Cost Calculator

This program calculates the total cost of a shopping cart containing two items.

Step 1: ItemToPurchase Class

- Attributes:

\* item\_name (string): Name of the item

\* item\_price (float): Price per unit of the item

\* item\_quantity (int): Number of units purchased

- Constructor:

\* Initializes attributes to default values:

\* item\_name = "none"

\* item\_price = 0

\* item\_quantity = 0

- Method:

\* print\_item\_cost(): Prints the item name, quantity, price per unit, and total cost in the format:

"Item Name Quantity @ $Price per Unit = $Total Cost"

'''

class ItemToPurchase():

def \_\_init\_\_(self, item\_name='none', item\_price = 0, item\_quantity = 0):

self.item\_name = item\_name

self.item\_price = item\_price

self.item\_quantity = item\_quantity

def total(self):

return self.item\_price \* self.item\_quantity

def print\_item\_cost(self):

total\_cost = self.item\_quantity \* self.item\_price

print(f'{self.item\_name} {self.item\_quantity} @ ${self.item\_price:.2f} per Unit = ${total\_cost:.2f}')

# Part 2: Main function

"""

Program Outline: Shopping Cart Cost Calculator

This program calculates the total cost of a shopping cart containing two items.

Step 2: Main Program

- Prompt the user to enter details (name, price, quantity) for two items.

- Create two `ItemToPurchase` objects using the provided information.

Step 3: Calculate and Display Total Cost

- Calculate the total cost by summing the cost of each item (price \* quantity).

- Print the total cost along with the individual item cost breakdowns.

"""

from ItemToPurchase import ItemToPurchase

def getNumItems():

# Get the number of items from the user with validation

while True:

try:

num\_items = int(input('How many items would you like to add to the cart? '))

if num\_items <= 0:

raise ValueError("The number of items must be a positive integer.")

break

except ValueError as e:

print(f"Invalid input: {e}. Please enter a valid positive integer.")

return num\_items

def getItems():

name = input('Please provide the name of your item: ')

# Input validation for price

while True:

try:

price = float(input('What was the price for the item? '))

if price < 0:

raise ValueError("Price cannot be negative.")

break

except ValueError as e:

print(f"Invalid input: {e}. Please enter a valid non-negative number.")

# Input validation for quantity

while True:

try:

quantity = int(input('How many items did you purchase? '))

if quantity < 0:

raise ValueError("Quantity cannot be negative.")

break

except ValueError as e:

print(f"Invalid input: {e}. Please enter a valid non-negative integer.")

return ItemToPurchase(name, price, quantity)

def main():

items = []

num\_items = getNumItems()

for \_ in range(num\_items):

item = getItems()

items.append(item)

item.print\_item\_cost()

total\_cost = sum(item.total() for item in items)

print(f'\nTotal Cost: ${total\_cost:.2f}')

if \_\_name\_\_ == '\_\_main\_\_':

main()

# Screenshots of Execution

## # Part 1:

A screenshot of a computer screen

Description automatically generated

# Results

The program was executed successfully with multiple test inputs to validate its functionality. For the item creation the attributes were correctly updated. The calculations for total cost were accurate and reflected the correct multiplication of price and quantity for each item. As displayed in the output, both the item details and the total cost were printed in a clear and organized manner.

For instance, in the test case where "Chocolate Chips" and "Bottled Water" were added to the cart, the program accurately computed the total cost as $13.00. The individual breakdowns—$3.00 for one unit of "Chocolate Chips" and $10.00 for ten units of "Bottled Water"—were both calculated and presented correctly.

These results demonstrate the program's reliability in handling inputs, performing arithmetic operations, and outputting formatted results. Thus, the program meets the requirements set forth in the project specification.

# Git Repository

Repository for this week’s assignment can be found at this [link](https://github.com/Mr-Abe/critical_thinking_assignments/tree/master/Module_3_Assignment).