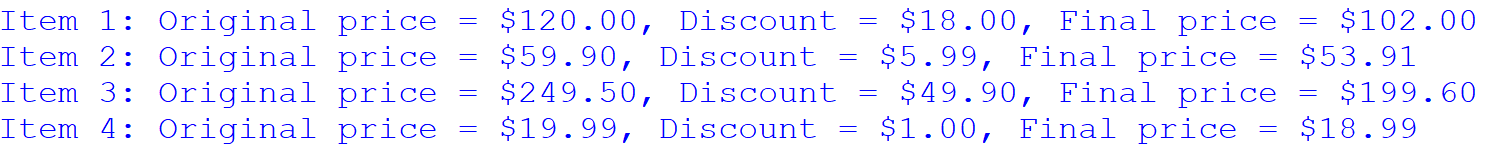
**Q1**

Write python code to calculate discounted prices for different items. Each item has a different discount rate as follows:

A screen shot of a price tag

AI-generated content may be incorrect.

**Sample run:**

****

**Suggested code:**

# Item 1

original\_price = 120.00 # e.g., the price of a shirt

discount\_rate = 0.15 # 15% discount

discount\_amount = original\_price \* discount\_rate

final\_price = original\_price - discount\_amount

print(f"Item 1: Original price = ${original\_price:.2f}, Discount = ${discount\_amount:.2f}, Final price = ${final\_price:.2f}")

# Item 2

original\_price = 59.90 # e.g., the price of a pair of socks

discount\_rate = 0.10 # 10% discount

discount\_amount = original\_price \* discount\_rate

final\_price = original\_price - discount\_amount

print(f"Item 2: Original price = ${original\_price:.2f}, Discount = ${discount\_amount:.2f}, Final price = ${final\_price:.2f}")

# Item 3

original\_price = 249.50 # e.g., the price of a bag

discount\_rate = 0.20 # 20% discount

discount\_amount = original\_price \* discount\_rate

final\_price = original\_price - discount\_amount

print(f"Item 3: Original price = ${original\_price:.2f}, Discount = ${discount\_amount:.2f}, Final price = ${final\_price:.2f}")

# Item 4

original\_price = 19.99 # e.g., the price of a keychain

discount\_rate = 0.05 # 5% discount

discount\_amount = original\_price \* discount\_rate

final\_price = original\_price - discount\_amount

print(f"Item 4: Original price = ${original\_price:.2f}, Discount = ${discount\_amount:.2f}, Final price = ${final\_price:.2f}")

**Q2. Reflection on code for Q1:**

Do you notice anything repetitive in your code?

What parts of your code are the same? What parts are different?

If you made a mistake in the logic, how many places do you need to fix it?

What if you had 100 items to calculate?

**Q3**

**Modify the discount price program stated in Q1 using function.**

**Suggested code:**

def apply\_discount(original\_price, discount\_rate):

discount\_amount = original\_price \* discount\_rate

final\_price = original\_price - discount\_amount

print(f"Original price = ${original\_price:.2f}, "

f"Discount = ${discount\_amount:.2f}, "

f"Final price = ${final\_price:.2f}")

# Reuse the function

apply\_discount(120.00, 0.15)

apply\_discount(59.90, 0.10)

apply\_discount(249.50, 0.20)

apply\_discount(19.99, 0.05)

**Q4 Reflection on code for Q3.**

Which version is shorter and easier to read?

Which version is easier to update if the logic changes?

Which version looks more organized and reusable?

If you were to test this logic, which would be easier to debug?

**Q5. Improve the discount price program in Q3 to show total price. The sample run is as follows:**

**A close-up of a white screen

AI-generated content may be incorrect.**

**Suggested code:**

# Function to apply discount and print item details

def apply\_discount(item\_label, original\_price, discount\_rate):

discount\_amount = original\_price \* discount\_rate

final\_price = original\_price - discount\_amount

print(f"{item\_label}: Original price = ${original\_price:.2f}, "

f"Discount = ${discount\_amount:.2f}, Final price = ${final\_price:.2f}")

return discount\_amount, final\_price

# New function to handle one item and update totals

def process\_item(item\_label, price, rate, total\_discount, total\_price):

d, p = apply\_discount(item\_label, price, rate)

total\_discount += d

total\_price += p

return total\_discount, total\_price

# Initialize totals

total\_discount = 0

total\_price = 0

# Process each item using the helper function

total\_discount, total\_price = process\_item("Item 1", 120.00, 0.15, total\_discount, total\_price)

total\_discount, total\_price = process\_item("Item 2", 59.90, 0.10, total\_discount, total\_price)

total\_discount, total\_price = process\_item("Item 3", 249.50, 0.20, total\_discount, total\_price)

total\_discount, total\_price = process\_item("Item 4", 19.99, 0.05, total\_discount, total\_price)

# Final totals (no summary function yet)

print(f"Total price to pay: ${total\_price:.2f}")

print(f"Total discount received: ${total\_discount:.2f}")

**Note to tutor: Revise the concept of Functions Calling Functions.**

**Q6 Enhance the program so that it no longer uses hardcoded values. Instead, use a while loop to repeatedly ask the user to input the item label, original price, and discount rate. The loop should continue until the user enters a special keyword like ‘End' to signal that there are no more items to process.**

**The sample run is as follows:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

***Optional: Error checking, case insensitivity***

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Suggested code:**

# Function to apply discount and print item details

def apply\_discount(item\_label, original\_price, discount\_rate):

discount\_amount = original\_price \* discount\_rate

final\_price = original\_price - discount\_amount

print(f"{item\_label}: Original price = ${original\_price:.2f}, "

f"Discount = ${discount\_amount:.2f}, Final price = ${final\_price:.2f}")

return discount\_amount, final\_price

# Function to handle one item and update totals

def process\_item(item\_label, price, rate, total\_discount, total\_price):

d, p = apply\_discount(item\_label, price, rate)

total\_discount += d

total\_price += p

return total\_discount, total\_price

# Initialize totals

total\_discount = 0

total\_price = 0

# Repeatedly read input from user

while True:

item\_label = input("Enter item name (or 'End' to finish): ")

if item\_label.lower() == "end":

break

# Prompt until valid price is entered

while True:

try:

original\_price = float(input("Enter original price: "))

if original\_price < 0:

print("Price cannot be negative.")

continue

break

except ValueError:

print("Invalid input. Please enter a numeric value for the price.")

# Prompt until valid discount is entered

while True:

try:

discount\_rate = float(input("Enter discount rate (e.g., 0.15 for 15%): "))

if discount\_rate < 0 or discount\_rate > 1:

print("Discount rate must be between 0 and 1.")

continue

break

except ValueError:

print("Invalid input. Please enter a numeric value for the discount rate.")

total\_discount, total\_price = process\_item(item\_label, original\_price, discount\_rate, total\_discount, total\_price)

# Final summary

print(f"\nTotal discount received: ${total\_discount:.2f}")

print(f"Total price to pay: ${total\_price:.2f}")