Chapter 9 – Product Class

Contents

Time required: 120 minutes

Please read the directions carefully before beginning the assignment.

- Comment your code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Pseudocode

1. Write pseudocode for the exercise first.
2. The pseudocode is your high-level planning and thinking through the solution for the program.
3. Your final solution may not be the same as your pseudocode. That is fine. I want evidence of planning before coding.

What I Want You to Do

Create a program using the following directions as a guideline. Your program does not have to look the same as the example run. I am looking for evidence of understanding of OOP concepts. Part of the evidence is commenting your code to show understanding of what you are doing.

Why I Want You to Do It

We want to develop the process of solving problems. It is not about getting the assignment done, but learning the process of planning and developing a program on your own.

One Step at a Time

Create the program one step at a time. Hard code and use print statements to test as you go.

1. Create the basic class. Put in the **__init__** and the method names.
2. Use **pass** in the method definitions. Create the skeleton of the class without coding it as shown in the scaffold code below.
3. Initialize an object with a name. Print the name to confirm it worked.
4. Get input from the user on the price. Print the name and price to confirm it worked.

Requirements

1. Create a Python class file named **product.py**
    a. **def __init__(self, name, price, quantity)**:
        i. When creating object, initialize the data attributes of the product. self._name = name
    b. Data Attributes
        i. **self._name** holds the product's name
        ii. **self._price** the regular price of the product
        iii. **self._quantity** the number of items of that product in stock
    c. Method **get_purchase():** get the price and quantity of items from the user.
        i. The regular price is charged for orders of less than 10 items.
            1. A 10% discount is applied for orders of between 10 and 99 items
            2. A 20% discount is applied for orders of 100 or more items.
        ii. Return a string displaying the purchase price

2. Create a Python program named: **product_program.py**
    a. Ask the user for a product name, price and quantity.
    b. Purchase a product at each level of discount.
    c. Display each individual sale.

Scaffold Code

The following code can be copied and pasted into your project. This will give you an outline to work with to complete your project.

product_class.py

```
"""
    Name: product_class.py
    Author:
    Created:
    Purpose: Calculate total sale with OOP
"""

class Product():
    # Initializes object with parameters
    def __init__(self, name, price, quantity):
        # Initialize private class variables from
        # incoming parameters
        self._name = name
        # TODO: Finish parameter assignments

    def calculate_purchase(self):
        """ Calculate the purchase price using internal data fields
        """
        pass    # pass is a placeholder until you put in code
        # TODO: Calculate purchase price

        # Return calculated purchase price as a string
        return item_price
```

product_program.py

```
"""
    Name: product_class.py
    Author:
    Created:
    Purpose: Calculate total sale with OOP
"""
import utils
import product_class
def main():
    print(utils.title("Welcome to the 4 Corners Store."))

    # TODO: Create menu loop for program for creation of multiple objects

    # TODO: Get input from user

    # Create a PurchasePrice() object
    product1 = product_class.Product(name, price, quantity)

    # Call calculate_purchase method from object
    purchase_price = product1.calculate_purchase()

    # TODO: Display result to screen


# If a standalone program, call the main function
# Else, use as a module
if __name__ == "__main__":
    main()
```

Example run:

```
+---------------------------------+
|  Welcome to the 4 Corners Store.  |
+---------------------------------+
Item name: Flour
Item price: 4.50
Quantity: 1

     Item name: Flour
    Item price: $4.50
      Quantity: 1
------------------------
Total Purchase: $4.50

Purchase another item? (y/n)y
Item name: Applies
Item price: .50
Quantity: 11
You have between 10 and 99 items.
10% Discount applies.

     Item name: Applies
    Item price: $0.50
      Quantity: 11
------------------------
Total Purchase: $4.95

Purchase another item? (y/n)y
Item name: Beans
Item price: .10
Quantity: 101
You have over 99 items.
20% Discount applies.

     Item name: Beans
    Item price: $0.10
      Quantity: 101
------------------------
Total Purchase: $8.08

Purchase another item? (y/n)n
```

Assignment Submission

1. Attach the pseudocode.
2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.