#Nathan Gregorian Bailey

#Section 0003

#Program 6

#Created on December 4th 2021

#Due December 11th 2021

Algorithm

1. Start.

2. Create class called Store.

2.1 Set constructor to have instance attributes of self, name, and location with default values of None and None respectively. Have self.name = name and self.location = location.

2.2 Create a setter method with attributes: self, storeName and storeLocation. Have self.name = storeName and self.location = storeLocation.

2.3 Creat display method with attribute: of self. Output the stores name and location

3. Create class called Cart that is inheriting the Store class.

3.1 Set constructor with instance attributes: of self, name, location, productName='', quantity =0, cart={'Milk':0,'Bread':0,'Egg':0,'Flour':0,'Oil':0,'Cheese':0},receipt=0. have a super().__init__(name,location). Set self.productName = productName, self.quantity = quantity, self.cart = cart, self.receipt = receipt.

3.2 Create the Add_item method with attributes self, product, quantity. have self.productName = product and self.quantity = quantity. If quantity entered is a negative number inform the user that they can only enter positive numbers. If self.productName is in self.cart increase the value for the key of self.cart[self.productName], otherwise inform the user that the store doesn't have the item they entered.

3.3 Create the Remove_item method with attributes: self, product, quantity. have self.procutName = product and self.quanitty = quantity. If quantity entered is a negative number inform the user that they can only enter positive numbers. Then check to see if self.productName is in self.cart AND that self.cart[self.productName] is greater than 0. If it is then subtract the self.quantity from self.cart[self.productName], and if after doing that

self.cart[self.priductname is less than zero set it to zero. Otherwise inform the user that the item they entered is not in their cart.

3.4 Create the receipt_setter method with attributes: self, receipt=0, prices = [2.50,1.98,.70,1.18,4.00,2.68]. have self.prices = prices, then for variables x and y in the enumeration of self.cart have the receipt add to it's value self.cart[y] multiplied by prices[x]. Have self.receipt = receipt and output self.receipt.

3.5 Create the display method with attribute: self. output that the items inside that cart are as follows. Then for x,y in the enumeration of self.cart if the value of self.cart[y] is greater than zero, output: "y with quantity of self.cart[y]".

4. Create a function called product_display.

4.1 Create the dictionary products = {'Milk':2.50,'Bread':1.98,'Egg':0.70,'Flour':1.18,'Oil':4.00,'Cheese':2.68} followed by outputing "Products as follows". Then for x,y in the enumeration of products, output "x : $ products[y].

5. Within the __main__, do the following:

5.1 Have variable b set to the class Store().

5.2 Ask the user to input the store's name and location and set them to store_name and store_location respectively.

5.3 Have b.setter() with attributes store_name and store_location.

5.4 Output b.display().

5.5 Output product_display().

5.6 Have variable c set to class Cart() with inherited attributes b.name and b.location.

5.7 Ask the user to to answer yes or no to if they want to add an item to their cart.

5.8 While True do the following.

5.8.1 If they entered yes then do the following.

5.8.1.A Ask the user to enter the name of the product they wish to add set to product_add.

5.8.1.B While True do the following.

5.8.1.B.1 Try the following.

5.8.1.B.1.1 Ask the user to enter the amount of product they want to add set to amount_add, and then break from the loop.

5.8.1.B.2 If the user did not enter an integer and exception is triggered for a ValueError and the user is informed to enter an integer.

5.8.1.C Have c.add_item() with attributes product_add and amount_add.

5.8.1.D Have c.display() and output the total receipt is $ c.receipt_setter().

5.8.1.E Ask the user if they want to add another product.

5.8.2 If they entered no then break from the loop.

5.8.3 If they did not enter yes or no, ask them to enter yes or no.

5.9 Ask the user if they would like to remove an item with yes or no.

5.10 While True do the following.

5.10.1 If they entered yes then do the following.

5.10.1.A Ask the user to enter the name of the product they wish to remove and set to product_remove.

5.10.1.B While True do the following.

5.10.1.B.1 Try the following.

5.10.1.B.1.1 Ask the user to enter the amount of product they want to remove set to amount_remove, and then break from the loop.

5.10.1.B.2 If the user did not enter an integer and exception is triggered for a ValueError and the user is informed to enter an integer.

5.10.1.C Have c.add_item() with attributes product_remove and

amount_remove.

5.10.1.D Have c.display() and output the total receipt is $ c.receipt_setter().

5.10.1.E Ask the user if they want to remove another product.

5.10.2 If they entered no then break from the loop.

5.10.3 If they did not enter yes or no, ask them to enter yes or no.

5.11 Ask the user to to answer yes or no to if they want to add an item to their cart and inform that this is their last chance to add items.

5.12 While True do the following.

5.12.1 If they entered yes then do the following.

5.12.1.A Ask the user to enter the name of the product they wish to add set to product_add.

5.12.1.B While True do the following.

5.12.1.B.1 Try the following.

5.12.1.B.1.1 Ask the user to enter the amount of product they want to add set to amount_add, and then break from the loop.

5.12.1.B.2 If the user did not enter an integer and exception is triggered for a ValueError and the user is informed to enter an integer.

5.12.1.C Have c.add_item() with attributes product_add and amount_add.

5.12.1.D Have c.display() and output the total receipt is $ c.receipt_setter().

5.12.1.E Ask the user if they want to add another product.

5.12.2 If they entered no then break from the loop.

5.12.3 If they did not enter yes or no, ask them to enter yes or no.

5.13 Output that you want to thank the user for shopping at the name of the store and

it's location.

5.14 c.display()

5.15 if c.receipt_setter() is less than or equal to zero do the following

5.15.1 Output that the cart is empty

5.16 Output that total of the users receipt which is $ c.receipt.setter().

6. End