# Module 4: Portfolio Milestone

Maurice Henning

Colorado State University Global

CSC500-1: Principals of Programming

Dr. Steven Evans

April 15, 2024

**Online Shopping Cart**

In this module, I will create an online shopping cart using classes, instances, and condition statements.

Step 1: The request was to create a class labeled with the provided specifications. By definition,

a class provides a means of bundling data and functionality. Creating a new class creates

a new type of object, allowing instances of that type to be made according to Abdelfattah (2023).

The class is to receive the label name ItemToPurchase. The class has 3 attributes and a default

constructor is defined. A default constructor is a unique function that gets called automatically

when an object of a class is created. The main purpose of a constructor is to initialize or assign

values to the data members of that class. It cannot return any value other than none as specified

by Loha (2022). The self attributes are assigned to none and zero. In this block of code, we

define a method to calculate and print these items in the specified format.

class ItemToPurchase:  
def\_init\_(self)  
self.item\_name= “none”  
self.item\_price=0  
self.item\_quantity = 0  
def print\_item\_cost(self)  
print (self.item\_name,self.item\_quantity,’@$’,end=’’)  
cost = self.item\_price,\*self.itemquantity  
print(cost)

Step 2: Requests the main section of the code to prompt the user for two items and create two of

the ItemToPurchase class. But first, we must create an ‘if’ condition statement for the name

variable for true false determination. Items 1 and 2 are associated with the ItemToPurchase class

. We execute a print statement asking the user to enter the item name, price, and quantity on

each line and this is performed for the 2nd item as well.

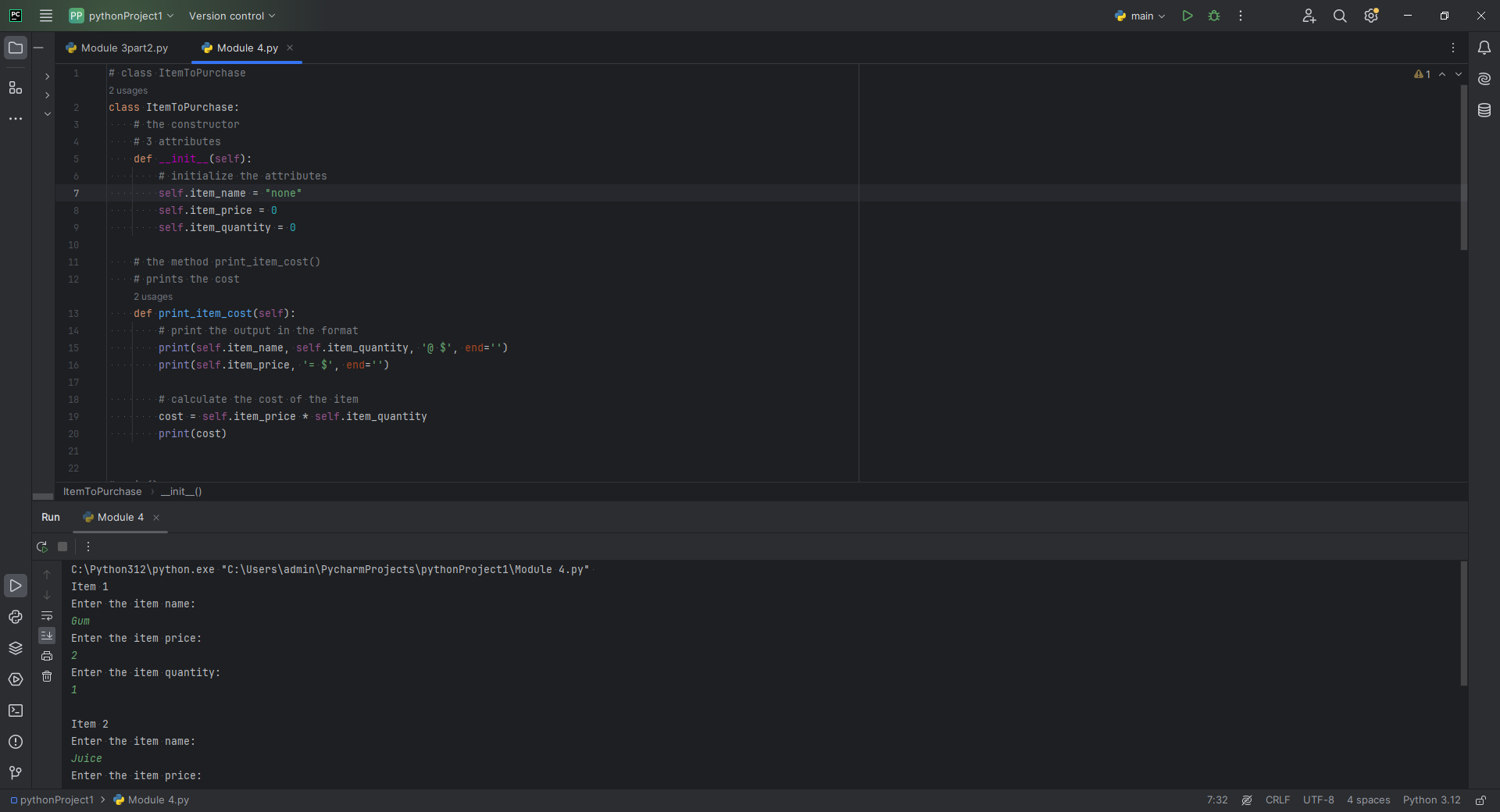
if \_\_name\_\_=="\_\_main\_\_":  
item1=ItemToPurchase()  
item2=ItemToPurchase()  
print('Item 1')  
item1.item\_name=input('Enter the item name:\n')  
item1.item\_price=int(input('Enter the item price:\n'))  
item1.item\_quantity=int(input('Enter the item quantity:\n'))  
item2.item\_name=input('Enter the item name:\n')  
item2.item\_price=int(input('Enter the item price:\n'))  
item2.item\_quantity=int(input('Enter the item quantity:\n'))

Step 3: Lastly I am requested to add the costs of the items together and output the total cost. We

create a simple print execution telling the program to print TOTAL COST of both items one and

two.

print ('TOTAL COST')  
item1.print\_item\_cost()   
item2.print\_item\_cost()  
print('\nTotal: $',end='')  
total\_cost=item1.item\_price\*item1.item\_quantity+item2.item\_price\*item2.item\_quantity print(total\_cost)

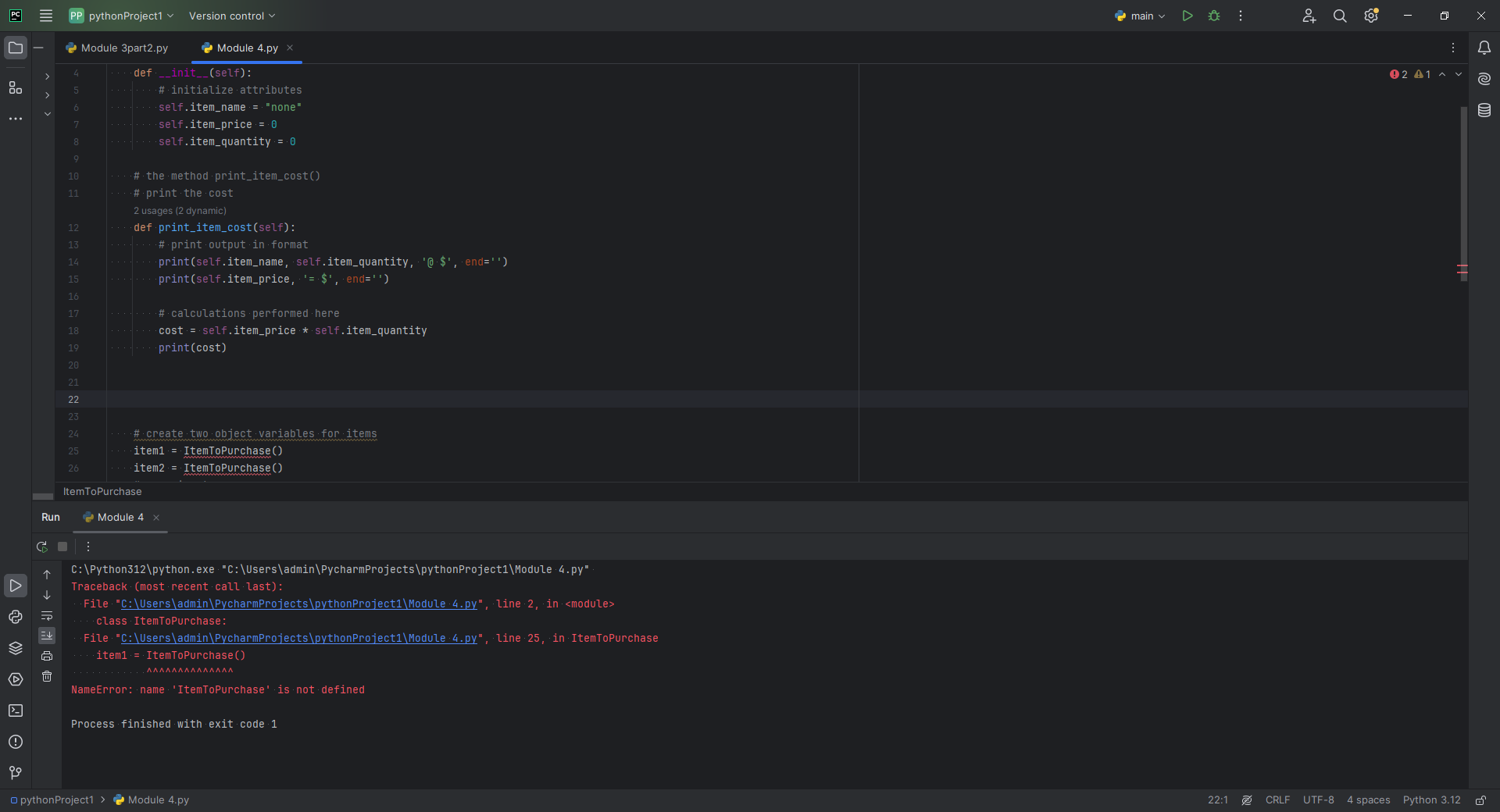


One of the challenges I encountered with creating this program was not creating a conditional if

statement which failed to define the ‘ItemToPurcchase condition statement. I discovered for this

to execute I would have to create the ‘if’ statement defining ItemTopurchase.

See below



**Pseudocode**

**Step 1:**

**1: CALL** item to purchase **2: SET initialization** for item attributes **3. SET** print item cost **4. SHOW** item name, and quantity in format **5. SHOW** price **6. CALCULATE** cost **7. SHOW** cost

**Step 2:**

**1: IF** name is true THEN proceed **2. SHOW** item 1 name **3. SHOW** item 1 price  
**4. SHOW** item 1 quantity **5. SHOW** item 2 name **6. SHOW** item 2 price **7. SHOW** item 2 quantity

**Step 3:**

**8. SHOW** total cost of items added **9. SHOW** total cost

**References**

Abdelfattah, N (2023). Understanding Classes in Python: From Templates to Objects   
<https://medium.com/@noransaber685/understanding-classes-in-python-from-templates-to-objects-8fa920aad3a1>

Lodha, K (2022). Constructor in Python  
<https://www.scaler.com/topics/constructor-in-python/>