Module 4 Portfolio Milestone 1

Online Shopping Cart: Steps 1, 2, and 3

Alexander Reichart-Anderson

Master of Science in Artificial Intelligence and Machine Learning, Colorado State University Global

CSC500-1: Principles of Programming

Dr. Steven Evans

February 9, 2024

Table of Contents

Shopping Cart Codebase

The online shopping cart, a ubiquitous feature of e-commerce, has revolutionized the way consumers interact with and purchase goods and services (Broder, 2025). Rayport & Jaworski's 2003 book *Introduction to e-Commerce* emphasizes how a good shopping cart directly impacts the user experience. This paper and the following milestones and code bases will outline the creation of an online shopping cart in the python programming language. The goal of the final project is to create a flexible and scalable code base that provides all the features of a Shopping Cart and delivers a great user experience.

Integrated Development Environment (IDE)

The following code bases, of this assignment, were developed with Python v. 3.10.4 (64-bit) in Visual Studio (VS) Code v. 1.96.4. These versions are the latest for the language and IDE to ensure functionality. If you were to recreate these code bases these versions should be used.

Code Repository

The code base for this portfolio project can be found in the publicly available Github repository here https://github.com/ara1data/CSC500_PrinciplesofProgramming/tree/main/PortfolioProject and in Appendix A (Reichart-Anderson, 2025). This Portfolio Project will be comprised of ten (10) main steps spread across three (3) submissions found in the following folders: Module4_Milestones1, Module6_Milestone2, and Module8_FinalSubmission. Each folder contains the code base (python file) and a markdown file with milestone instructions. This milestone, Milestone 1, can be found in the *Module_4_Milestone.py* python file and seen in Github in Appendix B.

Step 1: ItemToPurchase Python Class

A Python class is like a blueprint for creating objects (Hijazi, 2024). It defines the data (attributes) and actions (methods) that objects of that class will have. A real-world analogy for classes is a cookie cutter (Liffiton & Sheese, 2018). The class is the cutter, and the objects are the cookies that are made with it. Each cookie has the same shape (defined by the cutter), but they might have different decorations (different data values).

The assignment prompt(s), found in Milestone 1, contained the requirement for the following attributes: item_name (string), item_price (float), and item_quantity (int). The requirements also call for a default constructor that initializes item's name = "none", item's price = 0, item's quantity = 0. Finally, the requirements call for the creation of a print_item_cost() method.

The code, when executed, will have an example output of “Bottled Water 10 @ $1 = $10”.

Class Code Base

Step 1’s code can be seen between lines 21 and 28 (Appendix C). Line 21 *class ItemToPurchase:* declares the start of the new class. Line 22 *def __init__(self, item_name="none", item_price=0.0, item_quantity=0)*: is the constructor that initializes the attributes of the class giving default values and data formats to each attributes. Lines 23-25 assign values to each variable once they’re passed through the class. Line 27 *def print_item_cost(self):* creates that method print_item_cost that, when called, will print the values as shown in line 28 *print(f"{self.item_name} {self.item_quantity} @ ${self.item_price} = ${self.item_price * self.item_quantity}")*. The *f* at the begining of the string, creating a formatted string, allows for the embedding of attributes (those defined in the constructor).

Step 2: Prompt User for Item Input

Adding items to a cart is a key feature that proves the functionality of the functionality of the shopping cart. For this milestone, users will be prompted to enter the attributes for each item in the IDE terminal. The milestone 1, step 2 requirements, require the code base to ask the user to input item name, item price, and item quantity in the terminal. This feature will rely on the attribute formats that were outlined in the ItemToPurchase class.

Prompt User Code Base

Step 2’s code can be seen between lines 48 and 60 (Appendix D). Line 48 *if __name__ == "__main__":* is a python construct that calls this block of code within this file. Lines 49 and 50 create new ItemToPurchase objects and assign them to the variables item1 and item2. As mentioned in Step 1, these objects are passed the default values from Line 22. The lines 52-55 and 57-60 are the lines that prompt the user to input values for item 1 and item 2.

Item prices and quantity are converted to floats and integers respectively with the *float()* and i*nt()* functions. When the python file is executed, because of line 48, these prompts will be delivered to the user.

Step 3: Output Total Cost

An online shopping cart should be able to print the total cost of the items in the shopping cart. This feature allows the user to see how much money will be spent before "checking out" and beginning the procurement process. The step 3 requirements are to add the costs of the two items together and output the total cost.

Output of Cost Code Base

Step 3's code can be seen between lines 75 and 80 (Appendix E). Line 75 *print("\nTOTAL COST")* create a new line with *\n* and prints the label TOTAL COST. Line 76 and 77 calls the print_item_cost() method on the item1 and item1 objects. As outlined in Step 1, the ItemToPurchase class, prints the item name, quantity, unit cost, and total cost for each item. Line 79 creates a new variable *total_cost* that calculates the total cost of the two items in the shopping cart. Line 80 prints the total_cost variable.

Conclusion & Next Steps

When the python file is executed in Visual Studio Code the user is prompted to input each attribute for item 1 and then each attribute for item 2 (Appendix F). Once the item quantity for item 2 is entered the total cost code is executed and presented back to the user. The total cost code shows one line for each item, via the print_item_cost() method outlined in the ItemToPurchase class. In addition, the Total Cost is presented to the user by adding the values of item 1's quantity multiplied by price and item 2's quantity multiplied by price.

In the next milestone and final submission additional features will be added to the Online Shopping Cart Code base. Milestone 2 will focus on creating and printing a ShoppingCart python class that captures attributes regarding the user; methods for adding, removing, and modifying items; and additional features. Finally, Milestone 2 will enhance the output of total costs to include additional attributes to the user.

References

Broder, C. (2025). Advanced Shopping Cart Page Design Secrets: Elevate Your Checkout. *ECM*. https://theecommmanager.com/ecommerce/shopping-cart-page/

Hijazi, H. (2024). Understanding Python Classes and Instances: Building the Blueprint for Objects. *Medium*. https://medium.com/@hihijazi/understanding-python-classes-and-instances-building-the-blueprint-for-objects-c50652f58188

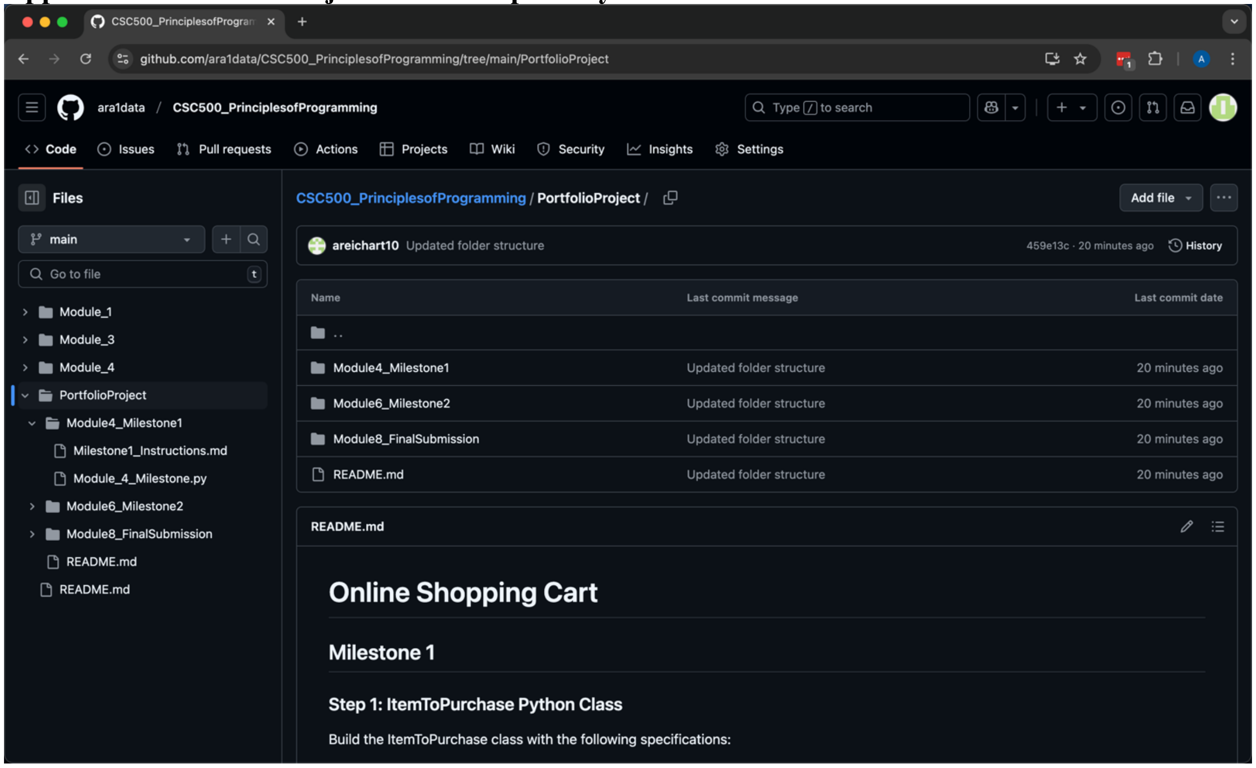
Liffiton, M. & Sheese, B. (2018). 11.2. Classes. *SnakeBear*. https://snakebear.science/11-Classes/classes.html

Rayport, J. & Jaworski, B. (2003). Introduction to e-Commerce. *McGraw-Hill*. https://dl.acm.org/doi/abs/10.5555/862014

Reichart-Anderson, A. (2025). Portfolio Project. *Github*. https://github.com/ara1data/CSC500_PrinciplesofProgramming/tree/main/PortfolioProject

Appendix

Appendix A: PortfolioProject Github Repository



Name	Last commit message	Last commit date
..		
Module4_Milestone1	Updated folder structure	20 minutes ago
Module6_Milestone2	Updated folder structure	20 minutes ago
Module8_FinalSubmission	Updated folder structure	20 minutes ago
README.md	Updated folder structure	20 minutes ago

README.md

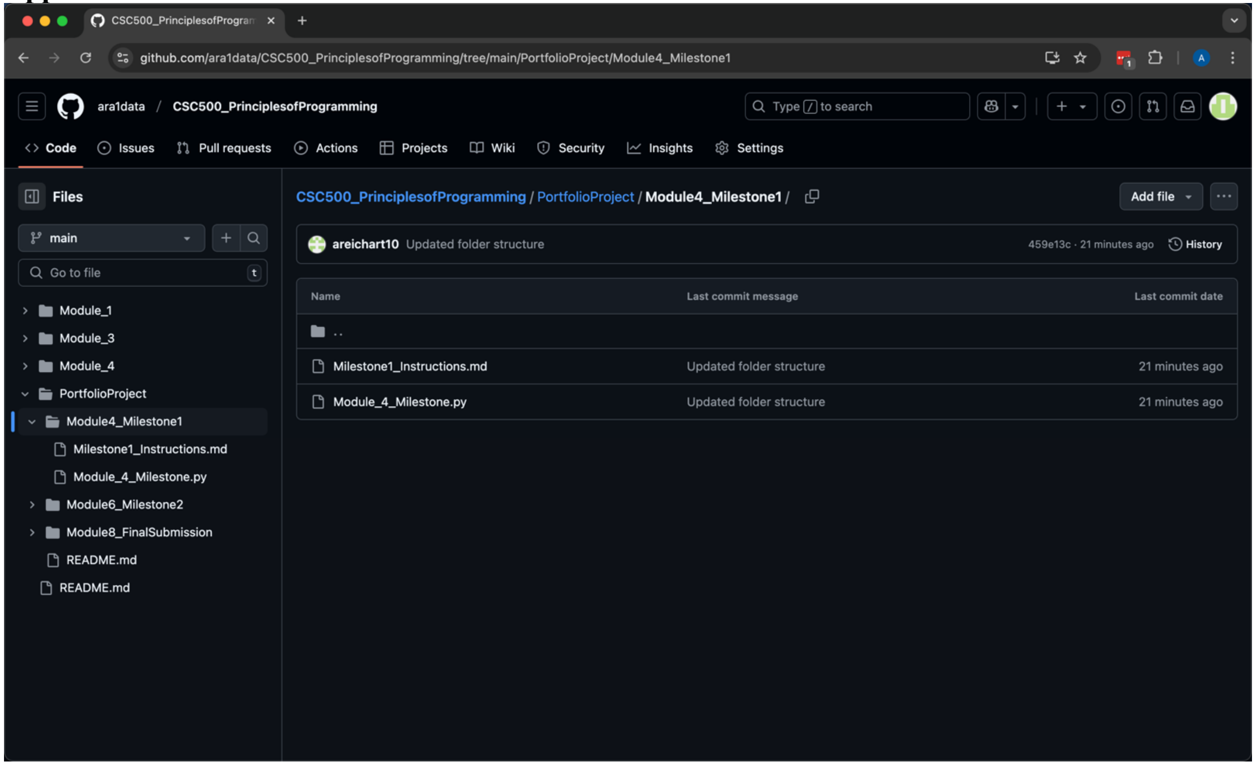
Online Shopping Cart

Milestone 1

Step 1: ItemToPurchase Python Class

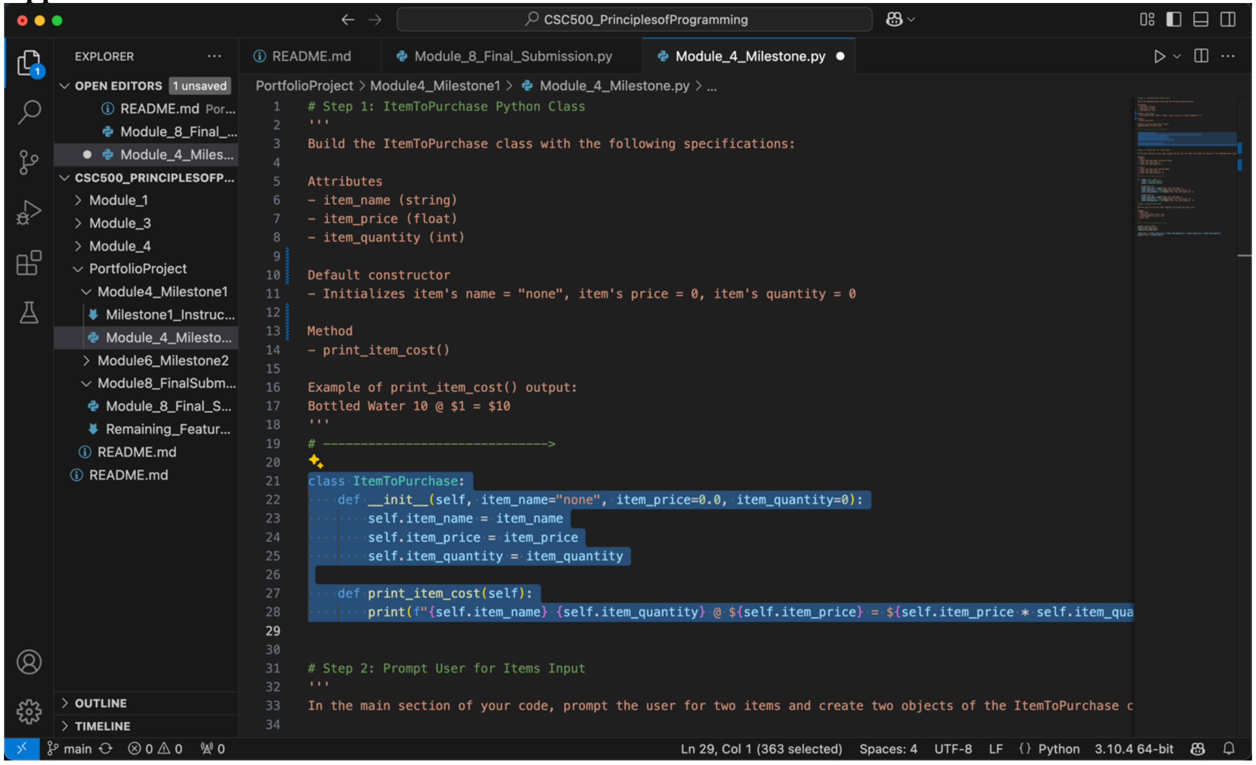
Build the ItemToPurchase class with the following specifications:

Appendix B: Milestone 1 Github Folder


github.com/ara1data/CSC500_PrinciplesofProgramming/tree/main/PortfolioProject/Module4_Milestone1
ara1data / CSC500_PrinciplesofProgramming
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Files
main
Go to file
Module_1
Module_3
Module_4
PortfolioProject
Module4_Milestone1
Milestone1_Instructions.md
Module_4_Milestone.py
Module6_Milestone2
Module8_FinalSubmission
README.md
README.md
CSC500_PrinciplesofProgramming / PortfolioProject / Module4_Milestone1 /
Add file
areichart10 Updated folder structure
459e13c · 21 minutes ago
History

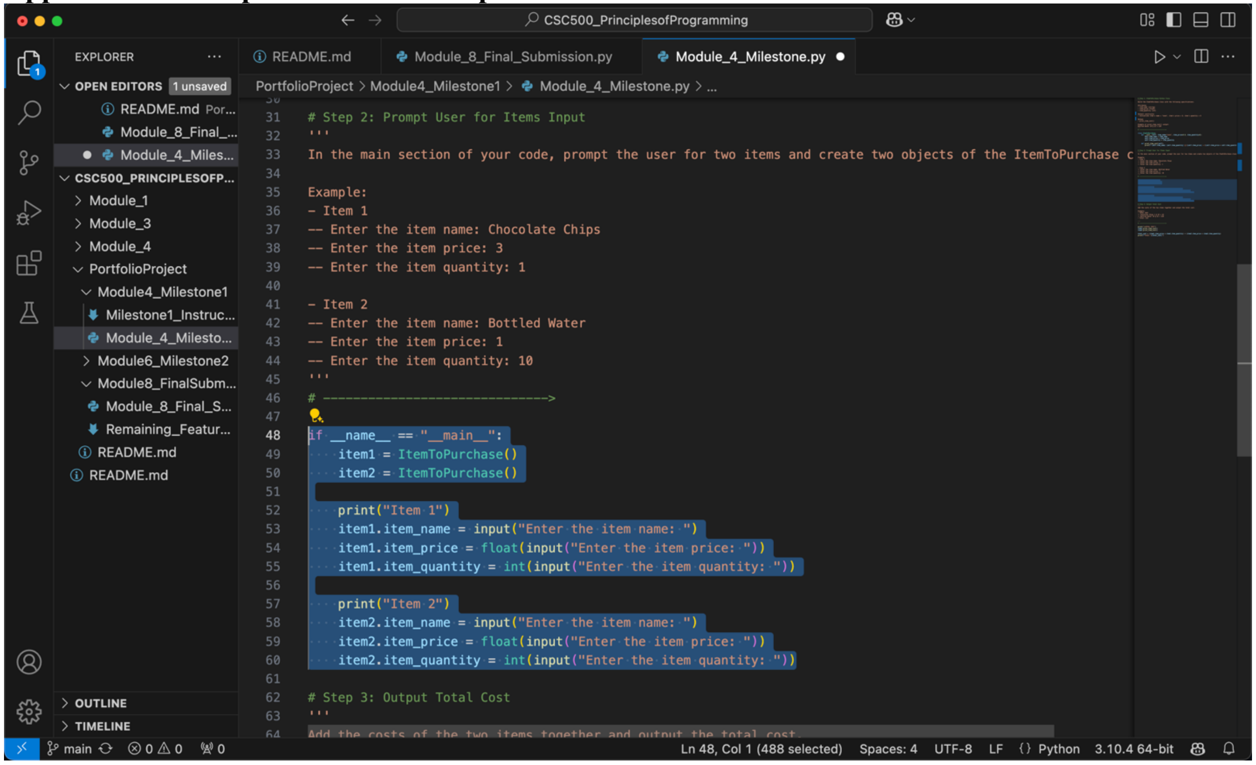
Name	Last commit message	Last commit date
..		
Milestone1_Instructions.md	Updated folder structure	21 minutes ago
Module_4_Milestone.py	Updated folder structure	21 minutes ago

Appendix C: ItemToPurchase Class



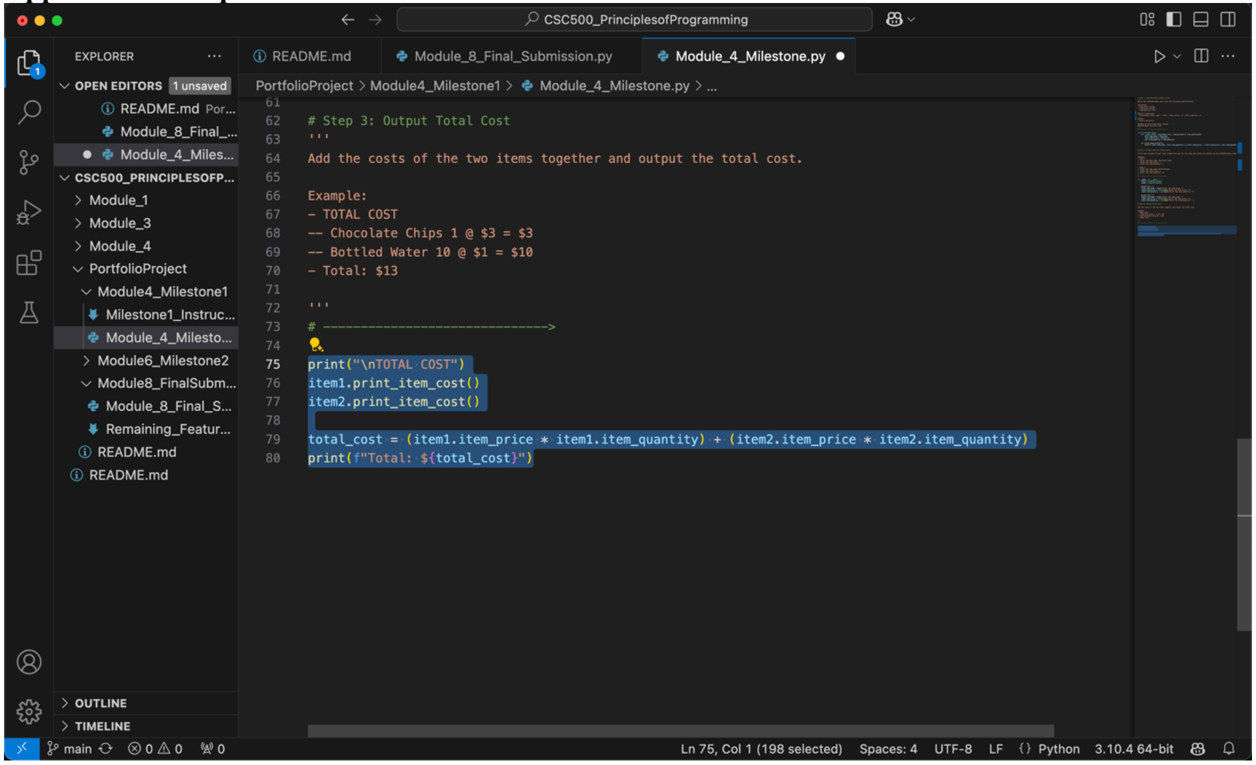
```
# Step 1: ItemToPurchase Python Class
'''
Build the ItemToPurchase class with the following specifications:

Attributes
- item_name (string)
- item_price (float)
- item_quantity (int)

Default constructor
- Initializes item's name = "none", item's price = 0, item's quantity = 0

Method
- print_item_cost()

Example of print_item_cost() output:
Bottled Water 10 @ $1 = $10
'''
# ---------------------------->

class ItemToPurchase:
    def __init__(self, item_name="none", item_price=0.0, item_quantity=0):
        self.item_name = item_name
        self.item_price = item_price
        self.item_quantity = item_quantity

    def print_item_cost(self):
        print(f"{self.item_name} {self.item_quantity} @ ${self.item_price} = ${self.item_price * self.item_qua


# Step 2: Prompt User for Items Input
'''
In the main section of your code, prompt the user for two items and create two objects of the ItemToPurchase c

```

Appendix D: Prompt User for Item Input


CSC500_PrinciplesofProgramming
EXPLORER
OPEN EDITORS 1 unsaved
README.md
Module_8_Final_Submission.py
Module_4_Milestone.py
PortfolioProject > Module4_Milestone1 > Module_4_Milestone.py > ...
OUTLINE
TIMELINE
main
Ln 48, Col 1 (488 selected) Spaces: 4 UTF-8 LF Python 3.10.4 64-bit

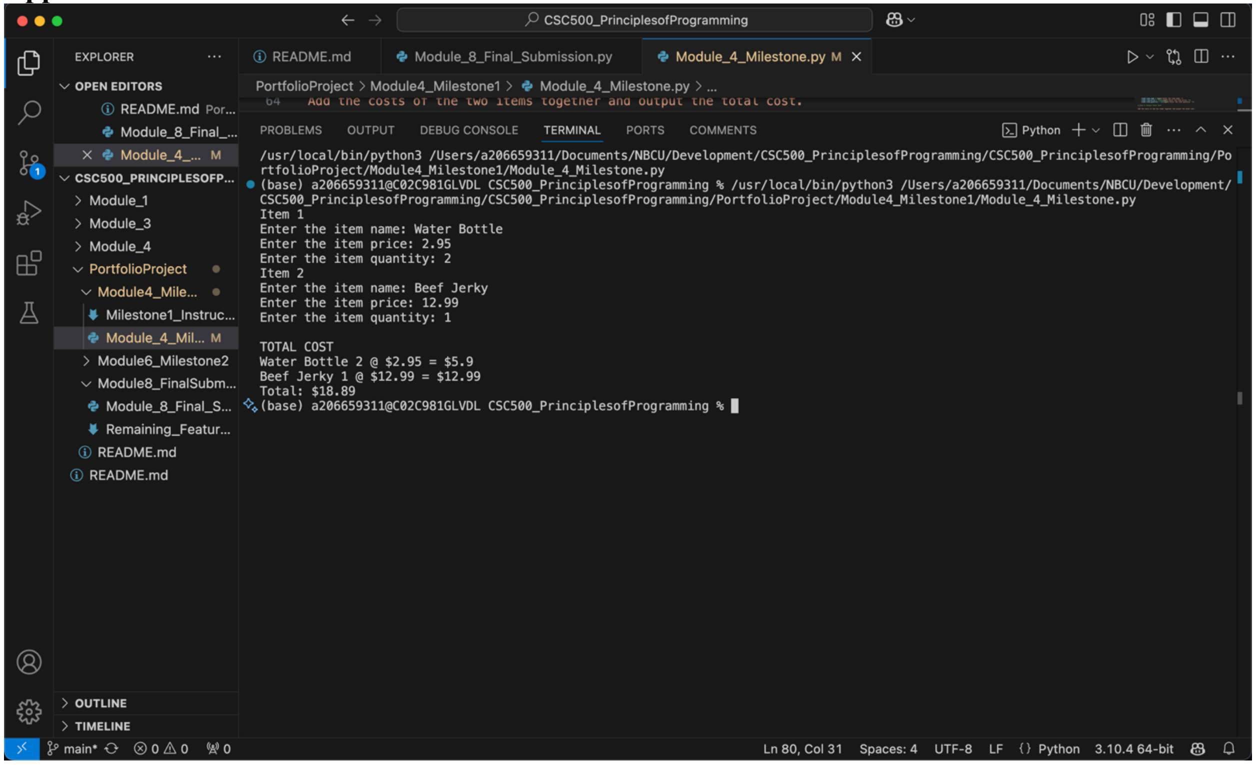
```
# Step 2: Prompt User for Items Input
'''
In the main section of your code, prompt the user for two items and create two objects of the ItemToPurchase c

Example:
- Item 1
-- Enter the item name: Chocolate Chips
-- Enter the item price: 3
-- Enter the item quantity: 1

- Item 2
-- Enter the item name: Bottled Water
-- Enter the item price: 1
-- Enter the item quantity: 10
'''
# ---------------------------->

if __name__ == "__main__":
    item1 = ItemToPurchase()
    item2 = ItemToPurchase()

    print("Item 1")
    item1.item_name = input("Enter the item name: ")
    item1.item_price = float(input("Enter the item price: "))
    item1.item_quantity = int(input("Enter the item quantity: "))

    print("Item 2")
    item2.item_name = input("Enter the item name: ")
    item2.item_price = float(input("Enter the item price: "))
    item2.item_quantity = int(input("Enter the item quantity: "))

# Step 3: Output Total Cost
'''
Add the costs of the two items together and output the total cost.
```

Appendix E: Output Total Cost



```
# Step 3: Output Total Cost
'''
Add the costs of the two items together and output the total cost.

Example:
- TOTAL COST
-- Chocolate Chips 1 @ $3 = $3
-- Bottled Water 10 @ $1 = $10
- Total: $13

'''
# ---------------------------->

print("\nTOTAL COST")
item1.print_item_cost()
item2.print_item_cost()

total_cost = (item1.item_price * item1.item_quantity) + (item2.item_price * item2.item_quantity)
print(f"Total: ${total_cost}")
```

Appendix F: Execution of code in terminal



```
/usr/local/bin/python3 /Users/a206659311/Documents/NBCU/Development/CSC500_PrinciplesofProgramming/CSC500_PrinciplesofProgramming/Po
rtfolioProject/Module4_Milestone1/Module_4_Milestone.py
(base) a206659311@C02C981GLVDL CSC500_PrinciplesofProgramming % /usr/local/bin/python3 /Users/a206659311/Documents/NBCU/Development/
CSC500_PrinciplesofProgramming/CSC500_PrinciplesofProgramming/PortfolioProject/Module4_Milestone1/Module_4_Milestone.py
Item 1
Enter the item name: Water Bottle
Enter the item price: 2.95
Enter the item quantity: 2
Item 2
Enter the item name: Beef Jerky
Enter the item price: 12.99
Enter the item quantity: 1

TOTAL COST
Water Bottle 2 @ $2.95 = $5.9
Beef Jerky 1 @ $12.99 = $12.99
Total: $18.89
(base) a206659311@C02C981GLVDL CSC500_PrinciplesofProgramming %
```