Module 4 Portfolio Milestone

Figure 1

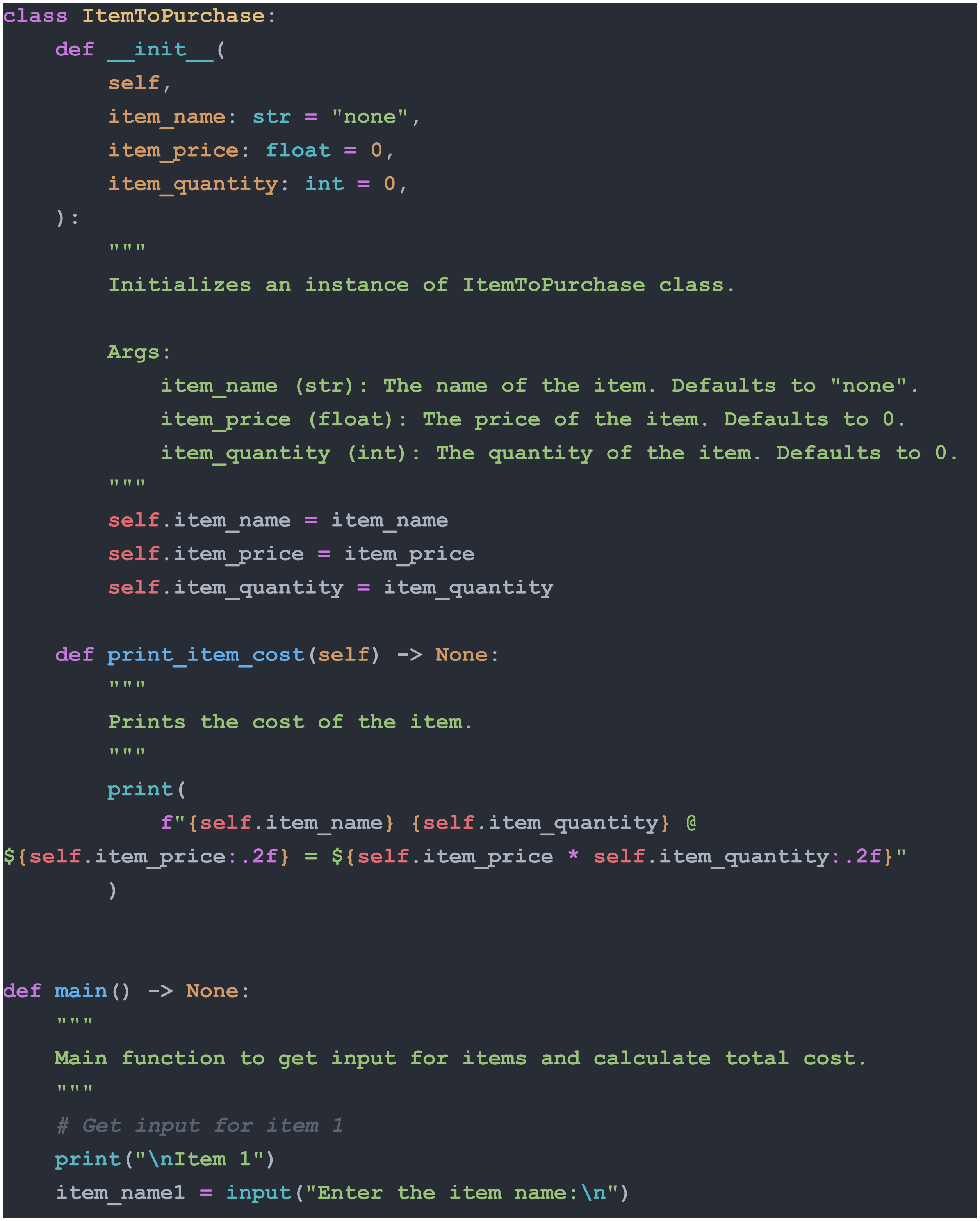
Pseudocode for Shopping Cart

```
START

CLASS ItemToPurchase

    FUNC __init__(item_name, item_price, item_quantity)

        SET self.item_name = item_name

        SET self.item_price = item_price

        SET self.item_quantity= item_quantity

    FUNC print_item_cost()

    PRINT "{item_name} {item_quantity} @ ${item_price} =

    ${item_price * item_quantity}"

FUNC main()

    PRINT "Item 1"

    PROMPT user for item_name1

    PROMPT user for item_price1

    PROMPT user for item_quantity1

    PRINT "Item 2"

    PROMPT user for item_name2

    PROMPT user for item_price2

    PROMPT user for item_quantity2


    PRINT "TOTAL COST"
    INSTANTIATE item1 = ItemToPurchase(item_name1, item_price1, item_quantity1)
    INSTANTIATE item1 = ItemToPurchase(item_name2, item_price2, item_quantity2)


    CALL item1.print_item_cost()
    CALL item2.print_item_cost()
    SET total = (item1.item_price * item1.item_quantity) + (item2.item_price * item2.item_quantity)
    PRINT “Total: “, total
IF __name__==’__main__’
    CALL main()
END
```

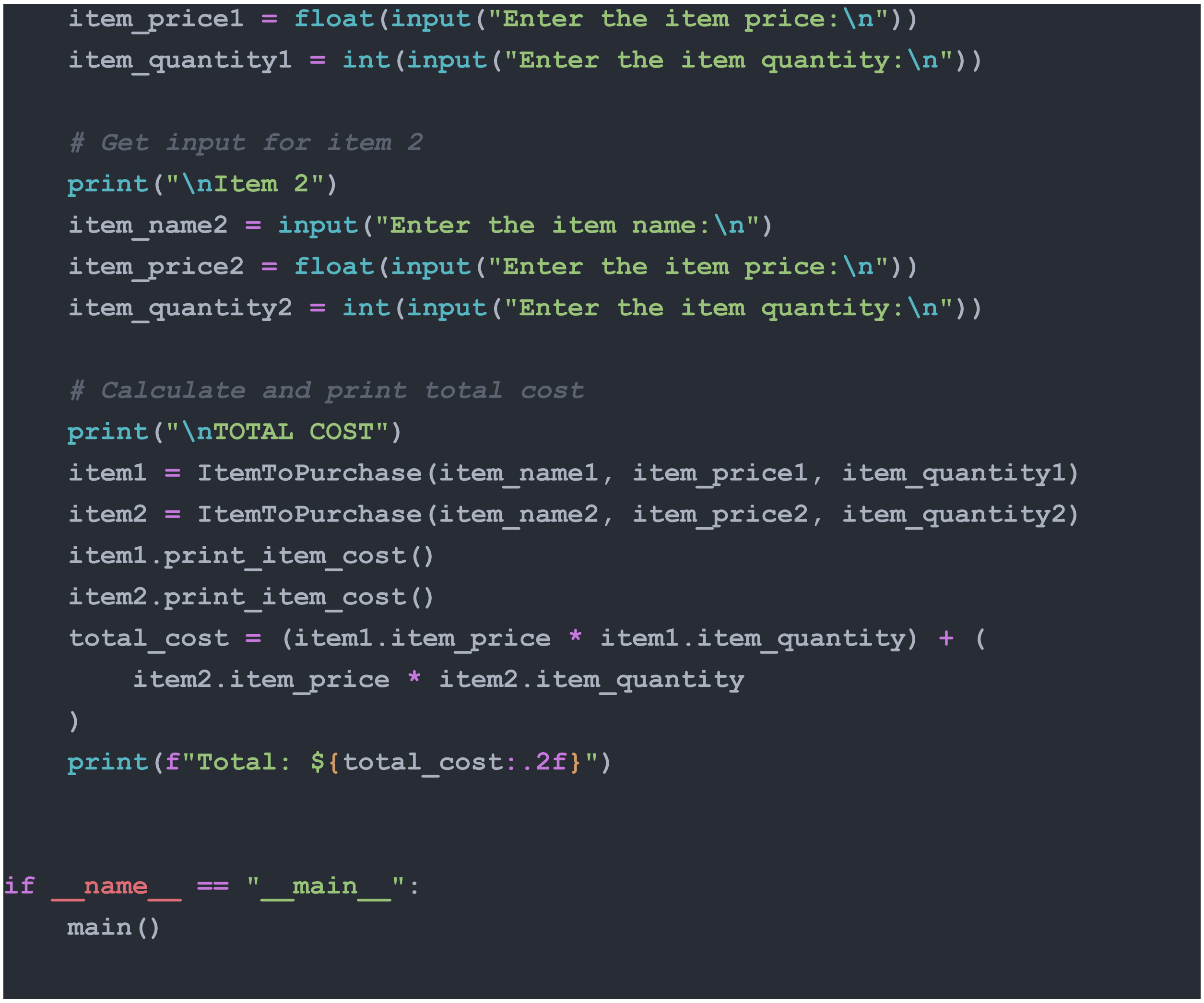
Note. This pseudocode illustrates a simple shopping cart algorithm that utilizes classes to create two objects, or items in this case, and initialize attributes such as item_name, item_price, and item_quantity based on user input. The shopping cart total is then displayed to the user along with the item’s name, price, and quantity.

Figure 2

Source Code for Shopping Cart



```python
class ItemToPurchase:
    def __init__(
        self,
        item_name: str = "none",
        item_price: float = 0,
        item_quantity: int = 0,
    ):
        """
        Initializes an instance of ItemToPurchase class.

        Args:
            item_name (str): The name of the item. Defaults to "none".
            item_price (float): The price of the item. Defaults to 0.
            item_quantity (int): The quantity of the item. Defaults to 0.
        """
        self.item_name = item_name
        self.item_price = item_price
        self.item_quantity = item_quantity

    def print_item_cost(self) -> None:
        """
        Prints the cost of the item.
        """
        print(
            f"{self.item_name} {self.item_quantity} @ ${self.item_price:.2f} = ${self.item_price * self.item_quantity:.2f}"
        )


def main() -> None:
    """
    Main function to get input for items and calculate total cost.
    """
    # Get input for item 1
    print("\nItem 1")
    item_name1 = input("Enter the item name:\n")
```

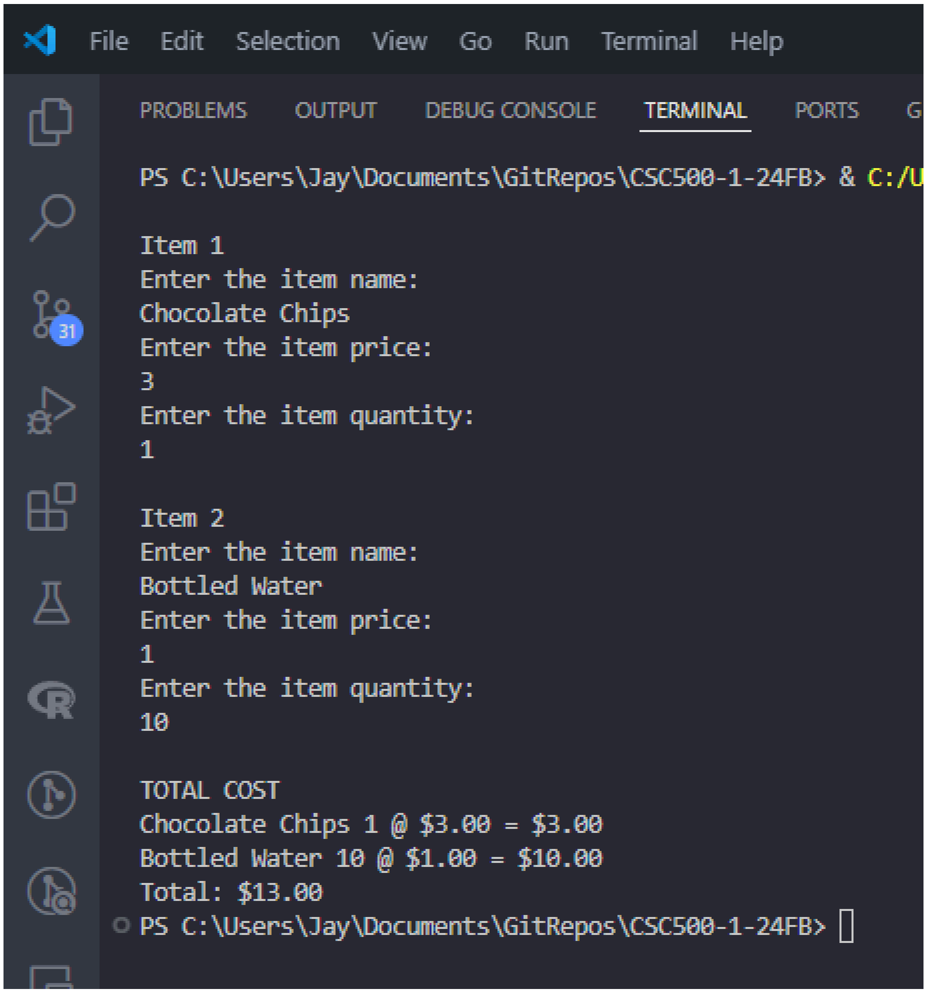


```
    item_price1 = float(input("Enter the item price:\n"))
    item_quantity1 = int(input("Enter the item quantity:\n"))

    # Get input for item 2
    print("\nItem 2")
    item_name2 = input("Enter the item name:\n")
    item_price2 = float(input("Enter the item price:\n"))
    item_quantity2 = int(input("Enter the item quantity:\n"))

    # Calculate and print total cost
    print("\nTOTAL COST")
    item1 = ItemToPurchase(item_name1, item_price1, item_quantity1)
    item2 = ItemToPurchase(item_name2, item_price2, item_quantity2)
    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:.2f}")


if __name__ == "__main__":
    main()
```

Note. This figure displays the source code used for a Python script that gets user input in order to instantiate the ItemToPurchase class two times for a total of two items. It then calculates the total cost based on the attributes item_price and item_quantity, and then displays the item names, quantities, prices, and total.

Figure 3

Execution and Testing for Restaurant Bill and Alarm Clock



```
PS C:\Users\Jay\Documents\GitRepos\CSC500-1-24FB> & C:/U

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

TOTAL COST
Chocolate Chips 1 @ $3.00 = $3.00
Bottled Water 10 @ $1.00 = $10.00
Total: $13.00
PS C:\Users\Jay\Documents\GitRepos\CSC500-1-24FB>
```

Note. Python output of a simple shopping cart that allows the user to provide two item names and then calculates the total cost of the items based on their quantity and price. According to the script, 1 Chocolate Chip at $3.00 each and 10 Bottled Water at $1.00 each will cost a total of $13.00.

References

Cline, J. T. [Jay4rmTheBay]. (2024). *CSC500-1-24FB* [Source code]. GitHub. https://github.com/Jay4rmTheBay/CSC500-1-24FB