**Module4 Critical Thinking Git Repo**

|  |
| --- |
|  |

**Portfolio Milestone source code**

**Problem statement:** Build the ItemToPurchase class with the given specifications

And produce the below output format

**Example:**

TOTAL COST

Chocolate Chips 1 @ $3 = $3

Bottled Water 10 @ $1 = $10

Total: $13

**Code:**

|  |
| --- |
| class ItemToPurchase :      item\_name: str      item\_price: float      item\_quantity: int        def \_\_init\_\_(self, item\_name: str, item\_price: float, item\_quantity: int):          self.item\_name = item\_name          self.item\_price= item\_price          self.item\_quantity= item\_quantity  ## Function to print the Total cost  def print\_item\_cost(items: list[ItemToPurchase]):      print()      print("###########")      print("TOTAL COST")      total\_cost = 0      for item in items:          total\_cost += round(float(item.item\_quantity \* item.item\_price), 2)          print(f"{item.item\_name} {item.item\_quantity} @ ${pricestr(item.item\_price)} = ${pricestr(round(float(item.item\_quantity \* item.item\_price), 2))}")        print(f"Total: ${pricestr(total\_cost)}")  def pricestr(v: float) -> str:      return f"{round(v , 2): .2f}"    if \_\_name\_\_ == "\_\_main\_\_":      items = []      for i in range(1, 3):          print(f"Item {i}")          item\_name = input("Enter the item name: ")          item\_price = round(float(input("Enter the item price: $")), 2)          item\_quantity = int(input("Enter the item quantity: "))            items.append(ItemToPurchase(item\_name=item\_name, item\_price=item\_price, item\_quantity=item\_quantity))      print\_item\_cost(items=items) |

**Portfolio Milestone Code execution:**

*Sample Input 1*

**A screenshot of a computer program

AI-generated content may be incorrect.**

*Sample Input 2*

**A screenshot of a computer program

AI-generated content may be incorrect.**