

Intro to programming  
Ezzat Mohamed  
0124544  
Vending machine

## **Introduction**

This vending machine is a python coding application of a real world vending system. However, This program is built using python coding, this coding focuses on creating friendly experience. This type of project is an example of how programming can be used in the real world and problem solving and can be used to solve tasks in the real world. This project is built to show how anything can be created and managed

## **Category:**

Firstly we should list the items, each item got its own ID, Name, stock and price. So the user can know the item there choosing Here is a example:

Item 1: Water, Priced at 1.50 with stock of 20

Item 2: Soda, Priced at 3.50 with stock of 7

This way the users are able and greeted with an organized way so they can see the products or items and choose whatever they want. All items are stored and kept in a key value in this dictionary.

For example if you press 1 for water so in the dictionary it is containing the details about the item like (id, name, price, stock)

```

1  # ----- THE VENDING MACHINE -----
2
3  #We should list the items with there prices to show each produt with its price in display
4  class VendingMachine:
5      def __init__(self):
6          """Initialize the vending machine with a list of items."""
7          self.items = {
8              1: {"id": 1, "category": "Drink", "name": "Water", "price": 1.50, "stock": 20},
9              2: {"id": 2, "category": "Drink", "name": "Soda", "price": 3.50, "stock": 7},
10             3: {"id": 3, "category": "Drink", "name": "Orange Juice", "price": 3.50, "stock": 8},
11             4: {"id": 4, "category": "Drink", "name": "Apple Juice", "price": 4.50, "stock": 10},
12             5: {"id": 5, "category": "Drink", "name": "Mango Juice", "price": 5.50, "stock": 7},
13             6: {"id": 6, "category": "Snack", "name": "Cheese Chips", "price": 4.50, "stock": 3},
14             7: {"id": 7, "category": "Snack", "name": "Chocolate", "price": 3.00, "stock": 10},
15             8: {"id": 8, "category": "Snack", "name": "Chocolate, With Peanuts", "price": 5.00, "stock": 10},
16             9: {"id": 9, "category": "Snack", "name": "Biscuit", "price": 5.50, "stock": 3},
17             10: {"id": 10, "category": "Snack", "name": "Salted Chips", "price": 6.00, "stock": 1},
18         }
19

```

## Displayed Items

Displayed items is a way that allows to display and show the available items in the vending machine, this makes it easier for users, while the items are displayed into two groups (Drinks and Snakes), However it uses (`_show_items_by_category`) to display the items for every category to make it easier and not complicated for users.

```

def display_items(self):
    """Show the items available in the vending machine grouped by category."""
    print("\n--- Welcome to the Vending Machine ---")
    self._show_items_by_category("Drink")
    self._show_items_by_category("Snack")

```

## Showing the items by category

This way to show category to users that can helps in many ways so This method filters and prints the items based on their category for example if i choose (item 1, Water) it prints for me the item's ID, name, price for every item i choose and it loops through (`self.items`) however it checks if each of the items matches the category and if the user chooses another ID the category would not appear for the user and the user should write or type

another ID so the category could appear and the items would appear for the user.

```
def _show_items_by_category(self, category):  
    """Display items of a specific category."""  
    print(f"\n{category}s:")  
    for item in self.items.values():  
        if item["category"] == category:  
            print(f"{item['id']}: {item['name']} - ${item['price']}")
```

## Allowing Users to Select items

Allowing users to select items is a way that makes the user type or write the item ID. This Method it makes the user choose the items that they want and add them to their 'cart'.

Before buying an item from the vending machine it asks the user to enter the item ID and checks if its available or invalid and it shows how many quantity is there so the user can add them to the cart and the user can choose multiple items by typing their ID and then it asks the user if they want to buy another items if not they would continue to the next step

Cart: It's the list where when you select an item its stored with its price, name, and quantity

```
def select_items(self):  
    """Allow users to select items and proceed to payment."""  
    cart = []  
    while True:  
        item_id = input("\nEnter item number to select (or 0 to finish): ")
```

## Checking if the Item ID is valid

This method makes sure that the user entered the correct item ID or number and if it's the wrong ID the vending machine would ask him to retype or rewrite the item ID or number again.

```
def _is_valid_item_id(self, item_id):  
    #Checks if the item ID is valid.  
    return item_id.isdigit() and int(item_id) in self.items
```

## Quantity of the items

By showing the users the quantity of the items it helps the users to specify in how many quantity they would like to get, at the same time the quantity must be more than 0 or equal to the available stock, By typing different ID that isn't shown in the list it won't process and the vending machine won't continue to the next step, the user must write or type the correct item ID so the vending machine get the users their items

If the item is out of stock the vending machine would ask the user to type a different iD so it can process and continue to the next step.

```
def _get_quantity(self, item):  
    #Prompt the user to enter the quantity of an item.  
    try:  
        quantity = int(input(f"Enter quantity for {item['name']} (Available: {item['stock']}): "))  
        if 0 < quantity <= item["stock"]:  
            return quantity  
        print("Invalid quantity. Try again.")  
        return 0  
    except ValueError:  
        print("Please enter a valid number.")  
        return 0
```

## Payment processing

Payment processing is a way to calculate the total cost of the items that has been selected and the quantity and the way it works is it calculates the total cost of every item that has been kept in the cart and the customer would enter the payment and then it calculates and gives back the users change and if the payment is not enough the vending machine would not process and the user would type the amount again.

```
def _process_payment(self, cart):
    #Calculate total cost and handle payment.
    total = sum(item["price"] * item["quantity"] for item in cart)
    print("\n--- Payment Summary ---")
    for item in cart:
        print(f"{item['name']} x{item['quantity']} - ${item['price']} * q")
    print(f"Total Amount: ${total:.2f}")
```

## Printing Receipt

After payment the customer would receive a receipt giving the customer a receipt with every thing that he bought like the name, quantity, and the cost of every item and it will even show the payment amount and the change that is left for the customer

```
def _print_receipt(self, cart, total, payment, change):
    #Generate and display a receipt for the user.
    print("\n--- Receipt ---")
    for item in cart:
        print(f"{item['name']} x{item['quantity']} - ${item['price']} *")
    print(f"Total: ${total:.2f}")
    print(f"Payment: ${payment:.2f}")
    print(f"Change: ${change:.2f}")
    print("\nThank you for using the vending machine!")
```

## OutPut

In the output it shows all the items in a type of list, it shows the Drinks and Snacks separate with there items ID and there price. It even asks for the quantity and then it asks for the payment method and at the end it shows the receipt.

```
Drinks:
1: Water - $1.5
2: Soda - $3.5
3: Orange Juice - $3.5
4: Apple Juice - $4.5
5: Mango Juice - $5.5

Snacks:
6: Cheese Chips - $4.5
7: Chocolate - $3.0
8: Chocolate, With Peanuts - $5.0
9: Biscuit - $5.5
10: Salted Chips - $6.0

Enter item number to select (or 0 to finish): 1
Enter quantity for Water (Available: 20): 3
Added 3x Water to your cart.
Do you want to select another item? (Yes/No): n

--- Payment Summary ---
Water x3 - $4.50
Total Amount: $4.50
Enter payment amount: $4.50
Payment successful! Your change: $0.00

--- Receipt ---
Water x3 - $4.50
Total: $4.50
Payment: $4.50
Change: $0.00

Thank you for using the vending machine!
```

## Conclusion

This vending machine is an example of how vending machines work and how easily you can use python to code that is easily made and it simulates a real world vending machine. It acts like a real vending machine and shows every single item. However, this python code has a practical skills

that helps in a strong understanding how software connects and interacts with real world scenario

## **Reference**

<https://www.tutorialspoint.com/how-do-you-code-a-vending-machine-in-python>

<https://copyassignment.com/vending-machine-with-python-code/>

<https://stackoverflow.com/questions/71955877/building-a-very-simple-vending-machine-in-python>