Pranjal Keshav Patil

Roll No.45

1. Create a simple program to manage products in a store using Python. Each product will have a name, price, and stock quantity. The program should allow adding products, updating stock, and viewing product details.

Class Definition

Create a Product class with the following:

• Attributes:

– name: The name of the product (string).

– price: The price of the product (float).

– stock: The quantity of the product in stock (integer).

• Methods:

– update_stock(quantity): Adds or removes the specified quantity from the product stock.

– **str**: Returns a string in the format: "Product: , Price: $, Stock: ".

Interactive Program

Create a simple program that:

1. Adds a new product.

2. Updates the stock of an existing product.

3. Displays the details of a product.

4. Exits the program.

Example Output

Step 1: Adding a Product

Enter product name: Laptop

Enter product price: 999.99

Enter product stock: 10

Product added successfully!

Step 2: Updating Stock

Enter product name to update: Laptop

Enter quantity to add/remove: -2

Stock updated successfully!

Step 3: Viewing Product Details

Enter product name: Laptop

Product: Laptop, Price: $999.99, Stock: 8

Step 4: Exiting the Program

Exiting the system. Goodbye!

```python
class Product:
    def __init__(self, name, price, stock):
        """Initializes the product with its name, price, and stock quantity."""
        self.name = name
        self.price = price
        self.stock = stock

    def update_stock(self, quantity):
        """Updates the stock quantity by adding or removing specified quantity."""
        self.stock += quantity
        if self.stock < 0:
            self.stock = 0

    def __str__(self):
        """Returns a string representation of the product details."""
        return f"Product: {self.name}, Price: ${self.price:.2f}, Stock: {self.stock}"

store_inventory = {}

def add_product(name, price, stock):
    new_product = Product(name, price, stock)
    store_inventory[name] = new_product
    print("Product added successfully!")

def update_stock(name, quantity):
    if name in store_inventory:
        product = store_inventory[name]
        product.update_stock(quantity)
        print("Stock updated successfully!")
    else:
        print(f"Product {name} not found.")

def view_product(name):
    if name in store_inventory:
        print(store_inventory[name])
    else:
        print(f"Product {name} not found.")

def main():
    while True:
        print("\nStore Management System:")
        print("1. Add Product")
        print("2. Update Stock")
        print("3. View Product Details")
        print("4. Exit")

        choice = input("Choose an option: ")

        if choice == "1":
            name = input("Enter product name: ")
            price = float(input("Enter product price: "))
            stock = int(input("Enter product stock: "))
            add_product(name, price, stock)

        elif choice == "2":
            name = input("Enter product name to update: ")
            quantity = int(input("Enter quantity to add/remove: "))
            update_stock(name, quantity)

        elif choice == "3":
            name = input("Enter product name to view details: ")
            view_product(name)

        elif choice == "4":
            print("Exiting the system. Goodbye!")
            break

        else:
            print("Invalid option. Please try again.")

main()
```

Output

```
Store Management System:
1. Add Product
2. Update Stock
3. View Product Details
4. Exit
Choose an option: 1
Enter product name: Laptop
Enter product price: 20000
Enter product stock: 4
Product added successfully!
Store Management System:
1. Add Product
2. Update Stock
3. View Product Details
4. Exit
Choose an option: 2
Enter product name to update: Laptop
Enter quantity to add/remove: +100
Stock updated successfully!
Store Management System:
1. Add Product
2. Update Stock
3. View Product Details
4. Exit
Choose an option: 3
Enter product name to view details: Laptop
Product: Laptop, Price: $20000.00, Stock: 104
Store Management System:
1. Add Product
2. Update Stock
```

```
3. View Product Details

4. Exit

Choose an option: 4

Exiting the system. Goodbye!
```