# Python Assignment no 1 by Parth Gawad (Roll no 62)

## 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.

First we will create class 'Product' in which the constructor method contains attributes like : name,price,stock of the product

```
class Product:
    # initializing constructor
    def __init__(self, name, price, stock):
        self.name = name
        self.price = price
        self.stock = stock

    def updatestock(self, quantity):
        self.stock += quantity  # updates the stock of the product

    #Returns the product info in one string
    def __str__(self):
        return f"Product Name: {self.name}, Price: ₹{self.price:.2f},
Stock: {self.stock}"
```

Now creating a class 'Store' to manage the product

```
class Store:
    def __init__(self):
        self.products = {}  # To store product information

    # adds product info in self.products variable
    def addPro(self, name, price, stock):
        if name in self.products:
            print(f"\nProduct {name} already exists in the list. Use
the Update option to modify.")
        else:
            self.products[name] = Product(name, price, stock)
            print(f"\nProduct {name} successfully added to the list.")

    # updates the numerical quantity of the specified product
    def updatestock(self, name, quantity):
        if name in self.products:
            self.products[name].updatestock(quantity)
            print(f"Stock for {name} updated by {quantity}")
        else:
            print(f"\nProduct {name} not found.")
```

```
    # display the specified product info
    def viewPro(self, name):
        if name in self.products:
            print(self.products[name])
        else:
            print(f"\nProduct {name} not found in the list.")

    # displays all product info
    def viewallPro(self):
        if not self.products:
            print("\n\nNo Products Present in the System.")
        else:
            print();
            for pro in self.products.values():
                print(pro)
```

Now in the main method, creating & calling the object of the classes & creating a basic interface for Users to perfrom operations

```
def main():
    store = Store() # object defination of the class
    # Interface handling
    while True:
        print("\nProduct Management System")
        print("1. Add Product\n2. Update Product Stock\n3. View
Product Info\n4. View All Products\n5. Exit")
        try:
            choice = int(input("Enter your choice: "))
        except ValueError:
            print("Invalid input! Please enter a number between 1 and
5.")
            continue

        if choice == 1:
            name = input("Enter the product name: ")
            try:
                price = float(input("Enter the product price: "))
                stock = int(input("Enter the product stock: "))
                store.addPro(name, price, stock)
            except ValueError:
                print("Invalid input! Price must be a number, and
stock must be an integer.")
        elif choice == 2:
            name = input("Enter the name of the product: ")
            try:
                quantity = int(input("Enter the quantity to update:
"))
                store.updatestock(name, quantity)
            except ValueError:
                print("Invalid input! Quantity must be an integer.")
```

```
        elif choice == 3:
            name = input("Enter the name of the product: ")
            store.viewPro(name)
        elif choice == 4:
            store.viewallPro()
        elif choice == 5:
            print("Exiting the program.")
            break
        else:
            print("Invalid input! Try again.")

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

```
Product Management System
1. Add Product
2. Update Product Stock
3. View Product Info
4. View All Products
5. Exit

Enter your choice:  1
Enter the product name:  5
Enter the product price:  213
Enter the product stock:  a

Invalid input! Price must be a number, and stock must be an integer.

Product Management System
1. Add Product
2. Update Product Stock
3. View Product Info
4. View All Products
5. Exit

Enter your choice:  5

Exiting the program.
```