

Problem Statement: Inventory Management System

You are tasked with designing an **Inventory Management System** for a small retail store. The system needs to manage information about products in the store, track product availability, and allow the store manager to add new products, search for products by name, and update the quantity of products when items are sold.

Requirements:

1. Classes to Implement:

- Product
- Inventory
- Main

2. Product Details:

- Each product has the following attributes:
 - name (String)
 - category (String)
 - price (double)
 - totalQuantity (int)
 - soldQuantity (int)
- When entering a product:
 - Ensure that the price is a positive value (greater than 0).
 - Ensure that soldQuantity cannot exceed totalQuantity (i.e., a product can't be sold more than what is available in stock).
 - If any of the input values are invalid, display an error message and ask the user to re-enter the information.

3. Commands/Functionalities:

- **Add Product:** Add a new product to the inventory with its details (name, category, price, total quantity).
- **Search Product:** Search for a product by name. If the product exists, display its details (name, category, price, total quantity, and available quantity).
- **Sell Product:** Update the sold quantity of a product. If there is sufficient stock available, reduce the available quantity and increase the sold quantity. If there isn't enough stock, display an error message.

4. Availability Calculation:

- The available quantity of each product should be calculated dynamically as $\text{totalQuantity} - \text{soldQuantity}$.
- When a product is sold, the available quantity should be updated accordingly.

5. Array and Command-line Arguments:

- The program should accept a command-line argument indicating how many products the store will manage initially.
- The details for each product should be entered manually by the user when the program runs.

6. Program Flow Example:

- The program should allow the store manager to perform multiple commands:
 - Add new products.
 - Search for products by name.
 - Sell products (if available).
 - Display available stock after each sale.

7. End Condition:

- The program should run continuously, allowing the user to input commands (add products, search products, sell products) until the user decides to exit the program.