****

**Assignment – Week 8 – Day 4**

You are tasked with building a simple inventory management system for a store. The program should allow the user to manage their inventory by adding new products, viewing all products, searching for a product by name or category, updating the price or quantity of a product, and removing a product from the inventory.

You can use a class called "Product" to represent each product in the inventory. The class should have the following attributes:

* **name** - the name of the product
* **category** - the category of the product
* **price** - the price of the product
* **quantity** - the quantity of the product in stock

The class should also have the following methods:

* **update\_price(new\_price)** - updates the price of the product to new\_price
* **update\_quantity(new\_quantity)** - updates the quantity of the product to new\_quantity
* **display\_product()** - displays information about the product (name, category, price, quantity)

**Requirements**

1. The program should allow the user to add new products to the inventory. When the user selects this option, they should be prompted to enter the name, category, price, and quantity of the new product.
2. The program should allow the user to view all products in the inventory. When the user selects this option, the program should display a list of all products in the inventory along with their name, category, price, and quantity.
3. The program should allow the user to search for a product by name or category. When the user selects this option, they should be prompted to enter the name or category of the product they are searching for. The program should then display a list of all products in the inventory that match the search criteria along with their name, category, price, and quantity.
4. The program should allow the user to update the price or quantity of a product. When the user selects this option, they should be prompted to enter the name of the product they want to update and the new price or quantity. The program should then update the price or quantity of the specified product.
5. The program should allow the user to remove a product from the inventory. When the user selects this option, they should be prompted to enter the name of the product they want to remove. The program should then remove the specified product from the inventory.
6. The program should have a menu that allows the user to select from the above options. The program should continue to display the menu until the user chooses to exit the program.
7. The program should store the inventory data in a list of Product objects.

**Library Management System**

**Description:**

You are tasked with creating a Library Management System program using Python. The program should allow librarians to keep track of the books in their library, their availability, and the borrowers. The program should present a menu to the user with several options to choose from.

**Requirements:**

1. The program should allow librarians to add a new book to the library by entering the book's title, author, and number of copies.
2. The program should allow librarians to search for a book in the library by entering a search term (e.g. title or author), and display a list of books that match the search term.
3. The program should allow librarians to display a list of all books in the library, including their titles, authors, and number of copies.
4. The program should allow librarians to display a list of books that are currently available for borrowing, including their titles, authors, and number of copies.
5. The program should allow librarians to display a list of books that are currently borrowed, including their titles, authors, and the names of the borrowers.
6. The program should allow borrowers to borrow a book from the library by entering the book's title and their name. If the book is available, the program should decrement the number of copies and add the borrower's name to a list of borrowers for that book.
7. The program should allow borrowers to return a book to the library by entering the book's title and their name. If the book is returned, the program should increment the number of copies and remove the borrower's name from the list of borrowers for that book.
8. The program should allow librarians to exit the program.

**Attributes:**

1. **title** - a string that represents the title of a book.
2. **author** - a string that represents the author of a book.
3. **num\_copies** - an integer that represents the number of copies of a book that are available in the library.
4. **borrowers** - a list of strings that represents the names of the borrowers who have borrowed a book.

**Methods:**

1. **add\_book()** - allows a librarian to add a new book to the library by entering the book's title, author, and number of copies.
2. **search\_book()** - allows a librarian to search for a book in the library by entering a search term (e.g. title or author), and display a list of books that match the search term.
3. **display\_books()** - allows a librarian to display a list of all books in the library, including their titles, authors, and number of copies.
4. **display\_available\_books()** - allows a librarian to display a list of books that are currently available for borrowing, including their titles, authors, and number of copies.
5. **display\_borrowed\_books()** - allows a librarian to display a list of books that are currently borrowed, including their titles, authors, and the names of the borrowers.
6. **borrow\_book()** - allows a borrower to borrow a book from the library by entering the book's title and their name. If the book is available, the program should decrement the number of copies and add the borrower's name to a list of borrowers for that book.
7. **return\_book()** - allows a borrower to return a book to the library by entering the book's title and their name. If the book is returned, the program should increment the number of copies and remove the borrower's name from the list of borrowers for that book.

Note: These are just suggestions for attributes and methods, and you may modify or add to them based on your implementation needs.

**Instructions:**

1. Create a Python program that implements the Library Management System described above.
2. Use a list to store information about books and their availability. You can assume that all book titles and author names are unique.
3. The program should present a menu to the user with the options listed above.
4. The program should handle user input and execute the appropriate function based on the user's choice.
5. The program should loop through the menu until the user chooses to exit.
6. The program should display appropriate messages to the user based on their input and the state of the library.
7. The program should be well-documented and easy to understand.

Test the program to make sure it works correctly and handles all possible input and edge cases.