

Task(10.2)

Inventory management

In this task, you will implement a simple inventory management system using C++ structs, vectors, and lambda functions. The system keeps track of items in an inventory, including their name, price, and quantity. The focus of this task is on using lambda expressions to update the inventory and perform calculations dynamically.

Key Features to Implement:

1. Struct Definition for Items:

- Define a struct called `Item` to represent an item in the inventory. The `Item` should have the following attributes:
 - `name (std::string)`: The name of the item.
 - `price (float)`: The price of the item.
 - `quantity (int)`: The quantity of the item available in stock.

2. Create an Inventory:

- Create a `std::vector` to store multiple `Item` objects, representing the inventory of a store. Initialize it with a few sample items such as "Laptop," "Smartphone," and "Tablet," each with a specific price and quantity.

3. Lambda Functions:

- **Update Item Price:** Write a lambda function `updatePrice` that captures the inventory by reference (`&inventory`). It takes an item's name and a new price as input, and updates the price of the specified item.
- **Check Stock Availability:** Create a lambda function `isOutOfStock` that captures the inventory by value. This function takes an item's name as input and returns a boolean indicating whether the item is out of stock (i.e., `quantity = 0`).
- **Calculate Total Inventory Value:** Implement a lambda `calculateTotalValue` that captures the inventory by reference. This function computes and returns the total value of the inventory, which is the sum of all items' prices multiplied by their quantities.

4. Main Program:

- Demonstrate the use of these lambdas by:
 - Updating the price of an item (e.g., "Smartphone").
 - Checking if an item (e.g., "Laptop") is out of stock.
 - Calculating the total value of the inventory.

Thank You