ITM SKILLS UNIVERSITY

C++ {Mini Project}

By Prof. Vinaya Mam

Group 5

Ishita Nakhawa

Project Title: Inventory Management System

Subtitle: A Digital Solution for Stock Tracking using C++

Technologies Used: C++, File Handling, STL

Problem Statement:

- Current Issue: Manual inventory tracking is prone to human error, leading to financial losses and stock-out situations.
- The Goal: To create a digital tool that automates the process of adding, viewing, and updating stock levels.
- Key Objective: Ensure data persistence so that stock information is saved even after the program closes.

C++ Concepts Applied

Classes & Objects: Used the Product class to encapsulate product data and behaviors.

Encapsulation: Private data members (id, name, quantity) protected from direct external access.

STL (Standard Template Library): Used std::vector for dynamic memory management of the product list.

File Handling: Used fstream (ifstream/ofstream) to read from and write to inventory.txt.

System Architecture & Logic Flow

Input Module: Captures user data via the input() method.

Processing Module: Handles logic for selling products and checking stock levels.

Storage Module: Manages the permanent storage of data in a text file.

Display Module: Formats and prints the inventory list to the console.

Startup: The system calls loadProducts() to pull existing data from inventory. txt.

Loop: A do-while loop keeps the program running until the user selects '0.'

Operations:

- **Add:** Appends a new object to the vector.
- **View:** Iterates through the vector using a for loop.
- **Sell:** Searches by ID and decrements quantity if stock is available.

Save: Every change triggers saveProducts() to update the file.

Module 1 – The Product Class (The Blueprint)

Data Encapsulation (Private Members)

To ensure data integrity, the product's core attributes are hidden from direct outside interference:

- int id: Unique identifier for the product.
- string name: Descriptive name.
- int quantity: Current stock level.
- float price: unit cost.

Public Interface (Methods)

These functions allow controlled interaction with the private data:

- **input()**: Standardizes how data is captured from the user.
- **display()**: Ensures a uniform tab-separated output format.
- **sell(int qty)**: The logic unit that updates stock levels.
- **getId() / getQuantity()**: Constant getter functions that provide read-only access to data.

Persistence Linkage

- **writeToFile() & readFromFile()**: These methods bridge the gap between the **object** and the **text file,** allowing the class to handle its own serialization.

ITM SKILLS UNIVERSITY

Thank you