

Module #5 Plan | CSE 310 – Applied Programming

Name	Date	Teacher
Ethan Trent	11/17/2025	Bro McGary

Software Description

SmallBiz is a console-based inventory management application for small businesses, developed in C++. The system will track products with SKU, name, quantity, price, and category information. It supports full CRUD operations (Create, Read, Update, Delete) for managing inventory items. The application will demonstrate C++ OOP principles through a product class hierarchy with inheritance and virtual functions, utilize STL containers (vector, map) for data management, implement file I/O for data persistence, and showcase dynamic memory management. The system includes features such as product search, sorted inventory display, and a user-friendly command-line interface with input validation.

Module

Mark an **X** next to the module you are planning

Module	Language	
Cloud Databases	Java	
Data Analysis	Kotlin	
Game Framework	R	
GIS Mapping	Erlang	
Mobile App	JavaScript	
Networking	C#	
Web Apps	TypeScript	
Language – C++	X	Rust
SQL Relational Databases	Choose Your Own Adventure	

Create a Schedule

Create a detailed schedule using the table below to complete your selected module during this Sprint. Include details such as what (task), when (time), where (location), and duration. You should also include time to work on your team project. You are expected to spend 16 hours every Sprint working on your individual module, team project, and other activities. Time spent on this individual module should be at least 10 hours.

First Week	Second Week
------------	-------------

	First Week	Second Week
Monday	2 hours - Setup development environment (Visual Studio/VS Code + MinGW), create GitHub repo, design class hierarchy (Product base class, PhysicalProduct/DigitalProduct derived classes). Location: Home/Computer Lab	2 hours - Implement file I/O functionality (save/load inventory to CSV), test data persistence, fix any bugs. Location: Home/Computer Lab
Tuesday	2 hours - Implement core Product class and derived classes with inheritance, add virtual functions for display and calculations. Location: Home/Computer Lab	2 hours - Create demo video (4-5 min walkthrough), finalize README.md with all sections, code review and documentation cleanup. Location: Home/Computer Lab
Wednesday	2 hours - Implement STL containers (vector/map) for inventory storage, create CRUD operation functions (Add, View, Edit, Remove). Location: Home/Computer Lab	1 hour - Final testing, polish UI/UX, prepare submission materials. Location: Home/Computer Lab
Thursday	2 hours - Implement search functionality, sorting algorithms, and menu-driven CLI interface with input validation. Location: Home/Computer Lab	1 hour - Team project work. Location: TBD
Friday	1 hour - Implement dynamic memory management (new/delete), test memory allocation/deallocation. Location: Home/Computer Lab	1 hour - Team project work. Location: TBD
Saturday	1 hour - Code review, add comments/documentation, test all features end-to-end. Location: Home/Computer Lab	1 hour - Team project work. Location: TBD

Identify Risks

Identify at least two risks that you feel will make it difficult to succeed in this module. Identify an action plan to overcome each of these risks.

	Risk	Action Plan
1	Difficulty understanding C++ OOP concepts (inheritance, virtual functions, polymorphism) and STL containers	Spend extra time reviewing C++ tutorials and documentation on cppreference.com. Practice with small example programs before implementing in main project. Seek help from instructor/TA during office hours or online resources like Stack Overflow. Break down complex concepts into smaller, manageable pieces.

	Risk	Action Plan
2	Memory management issues (memory leaks, segmentation faults) when using dynamic memory allocation with new/delete	Use smart pointers (unique_ptr, shared_ptr) where possible to reduce manual memory management. Test memory allocation/deallocation in isolated test cases before integrating. Use debugging tools (Valgrind on Linux, Visual Studio debugger on Windows) to detect memory leaks. Follow RAII principles and ensure every new has a corresponding delete.