COSC2081 Programming 1

A. Overview of the Assignment

This project is a group programming assignment where the main goal is to develop a Store Order Management System for a technology store with order management, admin and customer functions, and product and order information. The assignment comprises of designing the system, coding it, and preparing a report and a video demonstration.

Link for full code example: https://github.com/Autopilot7/STORE-ORDER-MANAGEMENT-SYSTEM

B. Steps to complete asignment

1. Requirements Understanding
2. Design Phase
3. Detailed Planning
4. Development Phase
5. Documentation and Reporting
6. Video Demonstration

C.1. Requirements Understanding

Project Details: You will develop a Store Order Management System for a technology store. The system should handle product management, member registrations, and order processing.

Functional Requirements: Features like member registration, product listing, order creation, admin functionalities, and data persistence must be implemented.

Technical Requirements: Input validation, data encapsulation, unique ID generation, and file-based data storage are mandatory.

OOP Design and Implementation:

- Design a class hierarchy for flexibility and maintenance ease.
- Problem Solving: Apply control statements, algorithms, data structures, etc., to solve given tasks.
- Report Writing: A 5-8 page report detailing your project.
- Class Diagram: Include a class diagram showing your system's structure.
- Video Demonstration: A short video explaining your analysis, design, implementation, and a demo of the system.

C.2. Design phase

For the "COSC2081 Programming 1" group assignment at RMIT University, which involves developing a text-based Order Management System, a detailed class diagram can be designed to represent the system's structure. Below is an example of classes and methods that might be included in such a system:

Classes and Methods:

1. **Product Class**
   - **Attributes:** `productId`, `productName`, `price`, `stockQuantity`
   - **Methods:**
     - `addProduct()`
     - `updateProduct()`
     - `deleteProduct()`
     - `getProductDetails()`
2. **Order Class**
   - **Attributes:** `orderId`, `orderDate`, `memberId`, `orderDetails`
   - **Methods:**
     - `createOrder()`
     - `cancelOrder()`
     - `updateOrder()`
     - `viewOrderDetails()`
3. **Member Class**
   - **Attributes:** `memberId`, `name`, `email`, `address`
   - **Methods:**

- `registerMember()`
- `updateMemberDetails()`
- `deleteMember()`
- `getMemberDetails()`

4. **Admin Class**
   - **Attributes:** Inherits or shares some attributes from `Member` (e.g., `adminId`, `name`, `email`)
   - **Methods:**
     - `addProduct()`
     - `removeProduct()`
     - `viewAllOrders()`
     - `updateOrderStatus()`
5. **OrderDetails Class**
   - **Attributes:** `orderId`, `productId`, `quantity`, `price`
   - **Methods:**
     - `addProductToOrder()`
     - `removeProductFromOrder()`
     - `updateQuantity()`
6. **Inventory Class**
   - **Attributes:** `products` (a list or collection of `Product` objects)
   - **Methods:**
     - `addStock()`
     - `reduceStock()`
     - `checkInventoryLevels()`

› Section hidden

[ ] ↳ *3 cells hidden*