Program 4: Object Oriented Design (Collectibles Store Simulation)

Overview

This program is designed to simulate an inventory management system that would be used by a collectibles store. The program will contain classes to represent domain concepts, container classes, and a main.cpp driver.

Domain Concepts:

Item Class: Item will serve as a base class for the three types of items to be tracked in this particular assignment: Coins, Comic books, and Sports cards. The derived classes will be labeled Coin, Comic, and SportsCard, respectively. These derived classes will contain Data Members specific for that item type, while Item will contain Data Members and Operations common to all item types. Each of the derived classes will also contain overridden comparison operators to assist with sorting.

Customer Class: Customer will contain all Data Members and Operations for customer-related actions and information such as purchasing and individual history of transactions. It will also contain a linked list of **Transaction Struct** that will store transaction history.

Store Class: Store will handle building the inventory and customer list by reading external text files. Store also handles any global actions such as displaying overall inventory, all customer history, and error messages.

InvTable Class: This class will contain the entire store inventory. The Store class will pass information about the items to this class, and using the hash function(s), will decide which buckets to pass the items into. Since no maximum number of items has been indicated for this project, the InvTable class will need to either employ open hashing (separate chaining) and/or be able to resize itself dynamically in case there are more items to be added than the size allocated to the Hash Table.

Factory Class: This class will take advantage of polymorphism in order to decide on which of derived classes to create. The class will actually pass on pointers to newly created objects of derived classes to the InvClass for sorting and storage.

BST Class: This class will act as a Binary Search Tree to contain all of the store's customers. Pointers to Customer objects are stored within a **BSTNode**. The Store class will pass information to this class in order to build the Binary Search Tree. Inorder traversal can be performed to visit all Customers in alphabetical order.

Main.cpp will begin and end execution of code. A Store object will be constructed, then initialized and its data members populated by reading an external file for the items, another for the customer list, and finally one for the sequence of commands to be processed.