Project 4: Associated Containers

Learning Goals:

- Familiarization with key/value association data structure usage and binary search tree concepts
- Familiarization and practice using the same key across associative containers to join the contents into a single view
- Familiarization and practice using the STL's map container interface
- Reinforce the similarities and differences between sequence and associative containers
- Reinforce reading persistent data from disk files and storing in memory resident data structures
- Reinforce modern C++ object-oriented programming techniques

Continuing our work for the Grocery Store, we are going to create an inventory and checkout system. We will have a system in place to process a customer's "cart" of desired purchases and update our store's inventory in real time. Each customer goes through the checkout line, is given a description for each item and its cost, and is given the total cost. When this happens, the system updates the inventory of the store to reflect the items that are no longer available, having already been purchased.

You are provided with starter code that works together with submissions from previous assignments to create the starting environment for this project.

1. **main.cpp -** Function main() orchestrates the flow of execution and tests your intermediate results along the way. File main.cpp is provided, requires no modifications, and will be overwritten during the grading process.The main flow can be summarized as:
    a. Load the store's inventory and a cart. The inventory and cart are both implemented using the OnlineOrder class, which contains a binary search tree where the key is the item's UPC and the quantity in stock is the value.
    b. Process customers' shopping carts by scanning all items in the basket, printing a receipt with amount due, and updating the store's inventory appropriately.
    c. Save the updated inventories to a new file using the OnlineOrder class's saveInventory method
2. **GroceryItem.hpp/GroceryItem.cpp** - Reuse the groceryitem class from previous assignments. Unless you find an error or omission in your code, these files should require no modification.
3. **GroceryItemDatabase.hpp/GroceryItemDatabase.cpp** - Reuse the GroceryItemDatabase class from your previous assignments, with modifications. You will modify the file to use an std::map as a container instead of the std::vector implementation that we already have. File GroceryItemDatabase.hpp is provided, requires no modifications, and will be overwritten during the grading process. You must modify and provide file GroceryItemDatabase.cpp.
4. **OnlineOrder.hpp/OnlineOrder.cpp** - contains a class that contains a map as a data class, where the key is a string representing the UPC of an item, and the value is an int

representing the amount remaining in the store. The header is provided to you, but you will have to define any methods in OnlineOrder.cpp

Files expected from you:
GroceryItem.cpp, GroceryItemDatabase.cpp, OnlineOrder.cpp, output.txt
and a copy each of the files below after completion:
_currentStoreInventory.dat, _currentShoppingCart.dat