Class: Store

string DisplayInventory()

Returns a list of all of the items that the store contains

map<int, string> Items()

Creates a list of items and quantities currently in the store

map<int, string> CartItems()

Returns a list of items and quantities in the Cart

void AddItemToCart(int item_id)

Called when item added to the ShoppingCart in order to decrease the quantity of that item from the store.

Also calls clone in order to create a copy of the item for the cart.

void RemoveItemFromCart(int item_id)

Called when a item is removed from ShoppingCart in order to increase the quantity of that item in the store

string DisplayCart()

Displays the items in the cart

double Checkout()

Calculates the cost for each item in the cart by multiplying the cost*the number of items in the cart* the tax.

Then overwrites the original file to say how much of each item is left in the inventory.

Resets the cart and returns the total cost of the items in the cart

void ClearCart()

Resets the items in the cart and the quantities to 0

Class: ShoppingCart

string DisplayCart()

Returns a list of all of the items currently in the cart

void Removeltem(Item * to_remove)

Decrements the quantity of the item in the cart by one if there are multiple of that item(calls DecreaseQuantity), otherwise if there is only one item,, it removes the item from the cart completely

void AddItem(Item * to_add)

If the item to be added is already in the cart, increases the quantity by one (calls IncreaseQuantity), otherwise, it adds one of the item to the list

void ClearCart()

Resets the items in the cart and the quantities to 0

vector<Item *> get_items()

Returns a vector of the items in the cart

Class: Item

Int get_id()
The ld of the item

int get_quantity()
How much of the item there is

double get_cost()

How much does it cost?

string get_type()
The type of item

void DecreaseQuantity(int amount)

Called by Removeltem to keep track of how many items are in the ShoppingCart
Also called by Store when AddItemtoCart is called, to keep track of the quantity of items left

void IncreaseQuantity(int amount)

Like DecreaseQuantity, used to keep track of how many items are in ShoppingCart and Store (items added to Store when removed from ShoppingCart, or added to ShoppingCart)

Item* Clone()

Used in AddItemToCart in order to create a copy of the item in the cart with the attributes of the item

string ToString() const

Class: TextUI

string MainMenu()
Creates a menu of options for the users

void RouteChoice(Store s, string choice)

int ItemMenu(Store s, map<int, string> items)