

## Ticket System Design

In designing the ticket system for In-n-Out, I decided to organize the program by grouping relevant data together in the form of classes. I initially thought to use structs for certain groups such as DateTime, but thought that a print function would be neat to have and also remembered that the time and date may have to change depending on things like daylight saving, so decided to use classes for all groups. The classes I created for the program were Item, Menu, Store, Date, Time, and Ticket.

All of the classes excluding the Ticket class were designed for use by the Ticket class. The Ticket class utilizes all other classes in order to print out the receipt using the methods included in these classes. To briefly describe each class and their functions, I will start the item class.

The item class includes functions like printItem(), setDescription(), setPrice(), setQuantity(), and getTotal(), which their purposes are as exactly as their name sounds. More specifically for the printItem function however, it will display the quantity, description, item price, and the total for that exact item taking into consideration the quantity and add-ons. The data stored in the item class include name, price, description, and quantity. It is to note here that all of the modifications to a given item are stored in the description attribute. I initially thought about creating individual objects for each item on the menu, but given that we do not know what is on the menu yet, I decided to make a template for any food items so that all of the necessary information are available to create any food item that In-N-Out may want to have. After In-n-Out decides which items they would like to have on their menu, they can create objects for each menu item and use them as references in copy constructors so that they do not have to manually create the items each time they take an order, and then use the setPrice() and setDescription() functions to add detail about the items in a given order.

A class that is directly related to the Item class was the Menu class. The Menu class consists of functions like printMenu(), addMenu(), and deleteMenu(), while having a list of pointers to items as its data. This list of items will include all of the items that In-N-Out decides to sell. It is to note here that every store class object will hold a pointer to the same Menu object so that every store has the same menu. If the company decides to raise the price of a certain item, they will simply have to change the price of the item in this list, which will then change that item in the menu for all stores. In the same manner, if the company decides to delete an item from the menu, they will simply have to delete it from the list of items in the main Menu object, which will then delete the item off of the menu of all stores. This implementation was made with the assumption that each store will have the same price for the same given items.

Next, the Store Class holds functions that include getTaxRate(), setTaxRate(), and printStoreInfo(). I did not include functions that change the address or the store number in the class because these information will not need to change. The printStoreInfo(), as the name suggests, will print all of the information related to the store except the taxRate, which will be

useful for printing on the receipt. The data that is included in the Store class are storeNumber, name, address, taxRate, and a pointer to a menu(the same pointer for all stores as mentioned above).

The final subclass is the DateTime class. This class is pretty straightforward, with the public interface including functions like printDateTime(), setHour(), setMinute(), and setSecond(). These modifier functions were included with things like daylight saving in mind. The data in this class are month, year, day, hour, minute, and second. One implementation that will not be shown in the header file is that this class will have logic to link all of the variables together. As time progresses each variable will increment accordingly, and once it has reached a certain threshold number (60 for minute, for example), it will increment the next unit.

Finally, I created a Ticket class, which is basically the receipt for each order. The Ticket class include functions getTaxAmount(), getFinalPrice(), and printTicket(). The data included in this class are a list of Item objects, a Store object, a DateTime object, and an orderNumber for that specific ticket. The printTicket() function will utilize all of the functions included in the various objects stored under this class to easily print the receipt for the customer. The DateTime class will print the date and time, the store class will print the store information along with the tax rate, and the Item object will print the sub total. The total will be calculated within this function using the getFinalPrice() function, the tax amount will be calculated using the getTaxAmount() function, and the orderNumber as well as the list of items will be accessed directly.