Ivan Solodyankin

Simon Mai

Dr. Kent Jones

CS 172

27 November 2018

Final Project Proposal

For our final project, we are going to do a simulation of a food court where a user can pick from a variety of places to eat from.  Currently there will be five venues to pick from: a burger joint, pizza place, taco stand, Chinese takeout, and an ice cream parlor.  Each place will allow the user to make a custom order from a variety of products from that place’s menu or if the user chooses, from multiple venues. The program will also track transactions made, placing them into separate text files. One of these text files will be a receipt while another one records the transaction history. From the transaction history, reports can be made about things like products that sold the most, which store sold the most, and report inventory.

There will be several classes. Each venue will have its own class that will inherit from a base venue class. Within each of the specialized venue classes, there will be main menu items and drink items. There will also be a class for transactions, which the venue classes can inherit from. Each store will have its own inventory text file as well as a transaction text file. The inventory file will be read from while the transaction file will be continuously updated and eventually be output on the transaction text file when the user is done with their eating spree.

Challenges that we anticipate include inheritance, specifically whether if each venue class will share inventory items. This would shorten the amount of time that would be required to make an inventory file for each class.  Tracking transaction history will also be another significant challenge, especially if the user decides to go to various places and buys multiple food and drink items from each individual place. Getting reports from the inventory presents another problem, since we will have to figure out how to read text files and write them out at the same time.

Project Design

|  |
| --- |
| store |
| * foodItem: string |
| * drinkItem: string |
| * foodItemPrice: double |
| * Vector <string> orders |
| * drinkItemPrice: double |
| * Store() |
| * Store(food: string, drink:string) |
| * getFood(): string |
| * getDrink(): string |
| * getfoodItemPrice() : double |
| * getdrinkItemPrice() : double |
| * makeOrder() : void |
| * addTip() : void |

Burger inherits from store

|  |
| --- |
| burger |
| * burger() |
| * bstore(food: string, drink:string) |
| * getFood(): string |
| * getDrink(): string |
| * foodItem(): string |
| * drinkItem(): string |

Taco Inherits from store

|  |
| --- |
| taco |
| * taco() |
| * tstore(food: string, drink:string) |
| * getFood(): string |
| * getDrink(): string |
| * foodItem(): string |
| * drinkItem(): string |

Pizza Inherits from store

|  |
| --- |
| pizza |
| * pizza() |
| * pstore(food: string, drink:string) |
| * getFood(): string |
| * getDrink(): string |
| * foodItem(): string |
| * drinkItem(): string |

Chinese Inherits from store

|  |
| --- |
| chinese |
| * chinese() |
| * cstore(food: string, drink:string) |
| * getFood(): string |
| * getDrink(): string |
| * foodItem(): string |
| * drinkItem(): string |

Icecream Inherits from store

|  |
| --- |
| icecream |
| * icecream() |
| * icstore(food: string, drink:string) |
| * getFood(): string |
| * getDrink(): string |
| * foodItem(): string |
| * drinkItem(): string |

//// Will need to make a get order factory function

Store \* GetOrder(string foodItem, string drinkItem )

{

   /////////////////// code

}

// overloading operator to read in orders from file

istream & operator>>( istream & is, store \* & c )

{

   string food, drink;

   is >> food >> drink;

   c = GetOrder(foodItem, drinkItem);

   return is;

};