Elisha Parslow, Justice Martinez, Elizabeth Young

FINAL PROJECT Requirements Specification

Switch-Case: Main menu options – all options leading to a function in the class pantryInventory

a) add inventory b) remove inventory c) list all complied inventory d) clear all inventory

Clear all inventory?

If Y/y – clear all of the text files then ask user to return to main menu or quit

If N/n – ask user to return to main menu or quit

Display sorted list of all items inventoried:

In text files, store inventory by adding the same word every time an inventory is added. When displaying inventory, count the # of words for that item, display the word and its count under its category

Ask user to return to main menu or quit

Switch-case: food categories

a) beverage

b) bread

c) breakfast 1. Select option

d) dessert 2. Enter food name

e) fruit 3. if exists –add/remove desired # to inventory

f) meat else- create new item with desired inventory # (in add option)

g) snack 4. Ask to inventory any other food in specific category

h) vegetable if yes- repeat steps 1-4, else move to step 5

5. Return to main menu or quit

Class pantryInventory{

private:

string item;

int count; //collected by counting strings of item

public:

pantryInventory(); //basic things in the pantry for us to implement the pantry functions without having to input inventory items multiple times

addItemCount(); //add a duplicate string to the text file for the # of inventory

addNewItem(); //add new item to food category

removeItem(); //remove item from food category

totalInventory(); //display the total inventory by categories, then foods in specific catagories with a number displayed after the item for the amount of that item, found by counting number of strings of that item

inventoryByCatagory(); //similar as total inventory but for a specific category, display at the end of a user adding inventory and removing inventory to that category

};

Problem Definition:

The program is expected to keep an inventory of items the user inputs in text files. The user has four options, to add or remove items, view the items or delete the entire pantry. In the File Input/Output, each food category will have its own text file to organize the storage. The inventory will be counted by the number of strings that item is written in the text document. In the view option, the program will count the number of strings, display the string of that item and then the count formatted as such:

Breakfast

Milk - 1

Cheese - 3

Eggs - 12

Dessert

Cake – 1

Ice-cream – 2

We don’t want to assume the user will remember if they put a capitalized item or not into the program so there will be a check to make all user input lower-case by *tolower*.