

# Assignment 2 –Design and Testing with Classes

## Goals-

Develop a class from given requirements

Implement that class using object-oriented programming techniques

You need to design, implement, and test a grocery list program. The program should display a list of items. The user must be able to enter and edit the items. They would need to save the list.

You will use a class for the items. The class should have data elements for the following information: item name, unit (i.e. can, or pounds, or ounces), number to buy, and unit price. Add other member variables you find necessary. You will implement creating the list, adding items, removing items, displaying the shopping list, reading a saved list from a file, and saving a list to a file. To add an item you should prompt the user to enter the name, unit of sale, the number needed, and the unit price.

You must create a design document. It can be submitted as part of the reflections but should provide some explanation of your choices. You need to consider the structure of your data. How are you ultimately going to represent them in your program? Figure that out, **before** you start coding. Then design the functions you will need. If you do the design properly the coding should be simple.

You will also need to provide a test plan. Since this is a program with input and output you should not need any driver functions. The test plan should provide coverage, i.e. several different items. You do NOT need to test for every possible item in a grocery store, just a reasonable number or varying lengths. How to do you handle spaces in names?

You can use any programming features covered in 161 or this course to date. Be creative. If you have more experience programming, remember this assignment gives more weight to the design and testing. If you make a more elaborate program then you have more design to document and explain, and probably more elaborate and systematic testing.

HINT: If you need a refresher on file operations you can wait for that part of the assignment until after you complete Lab 3.

## Grading:

- programming style and documentation (10%)
- create the list (20%)
- add and remove an item to the list (25%)
- display the list to the screen (5%)
- save the list to a file (10%)
- read a list from a file (10%)
- Display the list to include the following: item name, unit or purchase, price per unit, number to buy, subtotal for that item. Then at the bottom of the display indicate the total cost for that trip to the store. (5%)
- reflections document to include the design description, test plan, test results, and comments about how you resolved problems during the assignment (15%)