

# Assignment 06

## CS 120: Software Design I

### Program 1: Shopping

You will be writing a program that keeps track of a shopping list. Your code for this program should be written in the `Shopping` class included in the template code. To get started, **download** the template code and include it in your Eclipse project by dragging and dropping the `.java` file into the `src/` folder of the Eclipse project.

This program should allow the user to develop a grocery shopping list. You may assume that the list will contain **no more than 25 unique items**. If the user attempts to enter an item after hitting the 25 unique item limit, then your program should print an error and prompt again for input.

The user will enter items one at a time. When the user is finished building the list, they will enter the word `Done`, and the program should display the shopping list (in any order).

If an item is listed multiple times, then the program should keep track of the quantity of that item to pick up at the store. If the user wishes to reduce the quantity of an item on the list, then they can add a dash (-) in front of the item. Your program should then update the item quantity appropriately. For comparison purposes, your program should convert any user input to all capital letters. (See the Programming Hints for an example).

As you are considering writing your program, it would be best to start with some high-level sketch of your code, such as an activity diagram. Additionally, start by considering subproblems of this work. For example, what are all the possible actions the user can take? How can you tell which action the user has taken?

Several example runs of this program are shown below (user input is **colored**). Make sure that your output **exactly** matches these examples (except for the sections determined by user input).

**Pay particular attention to the linebreaks!**

```
Enter an item: milk
Enter an item: apple
Enter an item: peanut butter
Enter an item: done

## There are 3 items in the list.
## 1 x MILK
## 1 x APPLE
## 1 x PEANUT BUTTER
```

```
Enter an item: Dog Food
Enter an item: Eggs
Enter an item: Pineapple
Enter an item: Ice cream
Enter an item: -dog food
Enter an item: Pineapple
Enter an item: -eggs
Enter an item: Done

## There are 2 items in the list.
## 2 x PINEAPPLE
## 1 x ICE CREAM
```

```
Enter an item: eggs
Enter an item: milk
Enter an item: Milk
Enter an item: -EGGS
Enter an item: DONE

## There is 1 item in the list.
## 2 x MILK
```

The twist with this project is the 25 **unique** items requirement. Notice, in the example below, that I am trying to add 26 unique items, so when I try to add the 26th ('z') an error is displayed, 'z' is not added to the list, and the prompt appears again (since I might decide to remove an item to make room). The program should allow the user to remove an item, then add another in its place (it does not need to be in the exact same location in the list). You must think about how to manage your array(s) such that the example below will work correctly and produce the correct shopping list. Notice also that the 25 unique items limit does **not** include the quantity of the items.

```
Enter an item: a
Enter an item: a
Enter an item: a
Enter an item: b
Enter an item: c
Enter an item: d
Enter an item: e
Enter an item: f
Enter an item: g
Enter an item: h
Enter an item: i
Enter an item: j
Enter an item: k
Enter an item: l
Enter an item: m
Enter an item: n
Enter an item: o
Enter an item: p
Enter an item: q
Enter an item: r
Enter an item: s
Enter an item: t
```

```
Enter an item: u
Enter an item: v
Enter an item: w
Enter an item: x
Enter an item: y
Enter an item: z
Error: List is full!
Enter an item: -j
Enter an item: z
Enter an item: z
Enter an item: done

## There are 25 items in the list.
## 3 x A
## 1 x B
## 1 x C
## 1 x D
## 1 x E
## 1 x F
## 1 x G
## 1 x H
## 1 x I
## 2 x Z
## 1 x K
## 1 x L
## 1 x M
## 1 x N
## 1 x O
## 1 x P
## 1 x Q
## 1 x R
## 1 x S
## 1 x T
## 1 x U
## 1 x V
## 1 x W
## 1 x X
## 1 x Y
```

Here is an example of how attempting to remove an item before it is added to the list should have no effect.

```
Enter an item: -bacon
Enter an item: -Bacon
Enter an item: BACON
Enter an item: Done

## There is 1 item in the list.
## 1 x BACON
```

The following is a high level checklist of requirements for your program:

- Your class is named **Shopping** and is placed in the project created for this assignment

- Your code is commented according to the guidelines in the Java Style Guide found on D2L
- Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- Your code fulfills the functionality outlined above
- You have not modified any of the other code given to you, **particularly the method headers.**