

CSCI 204 – Introduction to Computer Science II

Lab 5: List ADT

1. Objectives

- Learn about the **List ADT**
- Implement the **List ADT** as an array.

2. Grocery Buddy

You work for a company which creates computer apps to help people in the everyday world. You are currently working on the GroceryBuddy app which keeps a shopping list. It displays the name, number, and picture of desired grocery items in a list. Items can be added and removed from the list. The desired quantity of items can be updated and item positions can be swapped. You have been assigned the task of implementing a standard **List ADT** so that the app will work.

3. List ADT

A list is a linear, ordered, indexed ADT. Like any ADT, a list has two sets of members, a collection of data and a set of operations defined on the data. The data held in a list can be of varying types. The set of operations (methods) supported by a list in general include the following:

- `constructor`: create an empty list
- `insert(item, index)`: insert a node with a given value at a location
- `delete(index)`: remove a node at a given location
- `len()`: return the size of the list

Some more elaborate lists may contain more operations. For this lab, you will also add the peek operation:

- `peek(index)`: return the item at that index

A list ADT can be implemented using different data structures. In this lab you will implement a list ADT using an **array**.

4. Getting started

This lab will use many files. If you haven't been yet creating separate directories for separate labs, now is a good time to start. Make this lab a directory named lab05 in your csci204 directory and move there. Copy the lab starter files from `~csci204/2017-fall/student/labs/lab05`. You should see four python files and many image (ppm) files.

```
main.py          graphics.py      GroceryBuddy.py listmain.py
avacados.ppm     bagels.ppm       bananas.ppm     beans.ppm ...
```

You will not need to modify any of these files. Rather all your lab work should be contained in two files, **arraylist.py** that has all the required elements of the class **List**, and **ListException.py** that has the class **ListException**. You must name these two files exactly as they are so the GroceryBuddy app will run as it is. Your task is to implement these two classes correctly in their respective files.

5. List ADT with an array

You must implement your list using an array. Therefore your list will need data attributes for the **array** of items, the **current size**, and the **current capacity**.

- **array:** is an array implemented as a Python List of size 2 and assign it to the array attribute. Here you are using the Python List to implement the array. You can use the indexing scheme of the Python list, but you have to manage the capacity of the list yourself. (See below.)
- **capacity:** represents the current size of your array underlying the List ADT. In the array above, the initial capacity is 2.
- **size:** represents the current number of elements in your List. When you first start out, the size is 0 (even though the underlying array has enough room to store 2 elements initially). The size of your List will change with each insert and delete operation, while the capacity will only change when the List needs to be expanded or shrunk.

NOTE: When the size is equal to capacity, it means that you've run out of room in your array to store more elements. At this point, your array should automatically double in size (along with your **capacity**) to clear up room for more elements.

Instead of simply appending more **None** to the end of your array using Python List operation, you will need to create a new list that is double the size of the old one, and copy the elements from your old array into the new one.

Remember to name your methods and fields so it is clear which ones should be accessed from outside. Those names that only get inside access start with an underscore. For example, your List will likely have **self._array** since it should not be accessed from outside the list.

6. Implement a ListException class

You are asked to implement a **ListException** class for the linked list. Your **ListException** class must inherit the **BaseException** class in order to become an exception. It will have very little in this class.

7. Implement a List class

Your List class **must** have the following standard methods:

- `__init__(self)`: creates an empty, unlimited list. The starting capacity should be 2, the starting size should be zero.
- `__len__(self)`: returns the size of the list.
- `insert(self, item, index)`: inserts an item at the given index. If the index is less than zero, insert at index 0. If the index is larger than or equal to the size, insert at the last spot. If the index is occupied, shove the item at the index (and all the items after that) towards the tail to make room. After you implement this method, run **main.py** and test the **Add** button.
- `delete(self, index)`: removes the item at the given index. If the index is illegal or unoccupied, do nothing. Do not return anything. After you implement this method, run **main.py** and test the **Remove** button.
- `peek(self, index)`: returns the item at the given index. If the index is illegal or unoccupied, raises a **ListException**. After you implement this method, run **main.py** and test the **Update** and **Swap** buttons.

After completing the implementation, you should be able to load and run the Python program **main.py**. When the program is running, try your shopping experience (!!) by adding any item that is represented by a graphics file, for example, eggs, beans, or others, or removing things that you already put in your shopping cart.

For your convenience, we also included a **listmain.py** file that you can test your list implementation without using the GroceryBuddy program.

8. Submitting your work

Congratulations! You've completed this lab. Please make sure submit your **arraylist.py** file on Moodle.

9. (Optional) To create more ppm images

To get ppm files, find some freely available image files online, save them in your directory, then open the **gif** or **jpg** file in Linux program **xv** and save as ppm raw file. If your addition is good and interesting, we will add it to our collection.