# Objectives

Practice Link list, List node, compareTo, user defined data type, interfaces

## Grocery List

Create an app to keep track of your groceries that you need to purchase. The information that you should keep track of each Item are the food, the price, and the expiration date of the food. In this app we want to create a dynamic list therefore LinkList must be used. Solutions without LinkList will not get any points.

# Required classes

* Item class
* ListNode class
* GroceryList class
* List interface
* Driver class

# Item Class

## This class must implement the comparable interface.

**Attributes**: food, price, and expiration date. Choose the proper data type for each of the instance variables. Food and expiration date are Strings and price is a double. Make sure to declare them all private.

**Methods**: constructor, getter, setter, equals, compareTo, and the toString. To create the equals method, consider that two items are the same if they have the same food and the same price. To implement the compareTo method, compare the two items based on the instance variable called food.

Refer to the PlayList example provided in the module for the week

# ListNode class

Attributes: each node has two attributes

1. Private Item i;
2. Private ListNode next

# Methods:

## Constructors (3)

* 1. public ListNode(Item i): initializes the instance variable i
  2. public ListNode() : empty body , no need to provide any code in this one
  3. public ListNode(Item i, ListNode next): initializes the instance variables i and the instance variable next

1. getter methods
   1. public Item getItem(): returns the instance variable item of the node
   2. public ListNode getNext() : returns the address part of the node

## setter methods

## public void setNext(ListNode next)

# List interface

## public void add(String food, double price, String expDate); public void add(int index, String food, double price, String expDate);

## public int indexOf(String food); returns the index of the food item in the list public void remove(String food); removes the food item from the list public int size(); returns the size of the list public String toString(); returns a string representing of all the food items in the list public Item get(int position); return the food item at the given position

## public Item getMostExpensive(); returns the most expensive food item in the list GroceryList implements List :

## all the methods in this class must use linklist concept

## public class GroceryList implements List {

## private ListNode head;

## public static int size = 0;

## //constructor

## public ItemList(){ head = null;

## }

## //add the food item to the end of the list

## **public void add(String food, double price, String expDate){}**

## //adds the food item at the given index

## **public void add(int index, String food, double price, String expDate){}**

## //returns the food item at the given index

## **public int indexOf(String food){}**

## //removes the food item from the list

## **public void remove(String food){}**

## //returns the size of the list

## **public int size(){} :** This method must loop through the link list and count the number of the nodes in the list

## //create a string from all the items in the list. Refer to the output if you need an example of what it looks like.

## **public String toString(){}**

## //returns the Item at the given index

## **public Item get (int pos){}** //return the Item at the given index

## //returns the Item with the highest cost. You need to go through the list of foods and see which one is the most expensive food in the list

## **public Item mostExpensive(){}**

# Driver class

Make sure that your code works with the following driver. Copy and paste this class to your file.

class Driver {

public static void main(String []args) {

GroceryList list = new GroceryList();

list.add("Bread", 5.99, "3/20/2022");

list.add("Milk", 3.99, "2/1/2002");

list.add("Chips", 2.99, "12/30/2025");

list.add("Rice", 35.50, "8/15/2030");

System.out.println("Here is the list of food items");

System.out.println(list);

System.out.println("Here is the most expensive item on the list");

System.out.println(list.mostExpensive());

System.out.println("Removing Milk from the list and adding a new expensive item on the list in the 2nd node");

list.remove("Milk");

list.add(1, "Truffle", 800, "4/20/2050");

System.out.println(list);

System.out.println("Testing the mostExpensive method to see what is the most expensive item now");

System.out.println(list.mostExpensive());

System.out.println("Testing the get method to get the item at the 3rd node");

System.out.println(list.get(2));

}

}

# Sample output

