# CST8130: Data Structures --- Assign #4

# Using ArrayList OR Dyamically Allocated Array with Hashing Data Structure

***DUE: Friday April 20 by 10PM SHARP!***

## Problem Description:

In this Assignment, we will re-write our Inventory program (Assignment #1 – dynamically allocated array or Assignment #2 - ArrayList) to handle inserts using a hash algorithm and a collision resolution algorithm.   Modify your code (or my solution) to do the following:

* Declare a dynamically allocated array called **inventory** of 100 **Item** objects OR an **ArrayList** of 100 Item objects. HINT: you should know how to do this assignment using both of these structures! But choose one
* When adding an item, use a hash algorithm to calculate the index of where to store the **Item** in the **inventory** array and a collision algorithm
  1. the hash algorithm should use the **itemCode**  – modulus 100 (so that you make sure the result is a number between 0 and 99).
  2. if there is already an **Item** in this index position – move to the next sequential element position in the array until you find an empty location (but not past 99! – in this case display error message – Item cannot be added)
* modify search for a **Item** in the array inventory to be efficient– and display the index of where it is found and the contents, or a message if it is not found
* buying and selling of Items should still work as in Assignment 1 and 2.

BONUS:

1. change the assignment to keep a LinkedList at each position in the array/ArrayList – and in event of a collision, add the Item to the Linked List (2 marks)
2. read and write with file – same files as used in Assignment 2 (2 marks)

***Submission:***

You must submit to the assignment link in Blackboard by the due date and time a zip file (named LastnameFirstNameAssign4) containing:

* all source code – ie .java files
* Your test plan

Failure to provide any of the above will have an effect on your grade for this assignment. Marking guide will be published shortly.