\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC249 data structure and alorithms**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAB 04 **ARRAYS**

# Objectives

- Understand how to build a class for a data structure

- Write code for the Array class

In this lab, we are going to build the Array class.

First, download the file array.py from Blackboard, which has the code of the Array class. This class is incomplete. Your task is to add code there to complete the class definition.

The Array class has the following data attributes:

self.items: A list that stores the data items. Initially, it is empty.

self.logicalSize: Number of data items in the array

self.capacity: Default capacity of the array. When an Array object is created, the default capacity is passed to the \_\_init\_\_ method as an argument. It should remain unchanged in the whole program. Whenever the array shrinks, it should never go below this default capacity.

self.fillValue: Value used to fill unoccupied slots in the array. Default value is None.

Please also note that the \_\_len\_\_ method returns the length of self.items, which is the current capacity of the array.

Some of the class methods are already defined. You need to write code for these four methods: insert, remove, clear and clone.

The insert method: This method has two parameters: index and newItem. It inserts newItem into the array at the position specified by index. If index is less than 0, insert newItem at position 0. If index is greater than the logical size of the array, insert newItem right after the last data item. For example, if currently there are 4 items in the array and index is 10, insert newItem at position 4. The first thing you should do in this method is to check whether the array is full. If it is full, call the \_\_grow method to increase its capacity before you insert newItem. Please read the section **Inserting an Item into an Array That Grows** on page 96-97 in the textbook before you write the code.

The remove method: This method has one parameter: index. It removes and returns the data item at position index from the array. If index is less than 0 or if it is greater than or equal to the logical size of the array, raise an IndexError and display the message "Array index out of bounds". After the data item is removed, compare the updated logical size and the current capacity of the array. If the logical size is less than or equal to one quarter of the current capacity, and the current capacity is at least twice the size of the default capacity, call the \_\_shrink method to reduce the size of the array. Don’t forget to return the removed data item. Please read the section **Removing an Item from an Array** on page 97-98 in the textbook before you write the code.

The clear method: This method has no parameter and no return value. It removes all data items from the array, resets its capacity to default capacity and fills every slot in the empty array with fillValue.

The clone method: This method has no parameter. It creates and returns a new Array object that is an exact copy of this Array object. For example, if this array is [0, 10, None, None, None], the new array will also be [0, 10, None, None, None].

The main method in the file array.py creates a few test cases. The code is already written. Do not change the code in the main method.

Expected output:

Initial array: [None, None, None, None, None]

Insert 3, 2, 1, 0: [3, 2, 1, 0, None]

Insert 77 at index 1: [3, 77, 2, 1, 0]

Insert 88 at index 2: [3, 77, 88, 2, 1, 0, None, None, None, None]

Insert 10 at index 15: [3, 77, 88, 2, 1, 0, 10, None, None, None]

Insert 66 at index -1: [66, 3, 77, 88, 2, 1, 0, 10, None, None]

Remove item at index 3: [66, 3, 77, 2, 1, 0, 10, None, None, None]

Remove item at index 0: [3, 77, 2, 1, 0, 10, None, None, None, None]

Remove item at index 0: [77, 2, 1, 0, 10, None, None, None, None, None]

Remove item at index 0: [2, 1, 0, 10, None, None, None, None, None, None]

Remove item at index 0: [1, 0, 10, None, None, None, None, None, None, None]

Remove item at index 0: [0, 10, None, None, None]

clone created: [0, 10, None, None, None]

clone cleared: [None, None, None, None, None]

After you have completed the definition of the Array class, submit the file array.py to Blackboard for credit.

# Grading rubric

The insert method [35 points]

The remove method [35 points]

The clear method [10 points]

The clone method [20 points]