**Arrays**

* array is a collection of items
* contiguous memory locations
* store items of the same type – homogeneous elements/items
* element can be uniquely identified by their index in the array

**Array’s size**

* fixed size -- can’t shrink it, can’t expand it

The shrinking will not work because array, when declared, it gets memory statically, and thus compiler is the only one to destroy it

The reason was that for expanding if we change the size we can’t be sure ( it’s not possible every time) that we get the next memory location to us as free

**Types of indexing in array:**

* 0 (zero-based indexing): The first element of the array is indexed by subscript of 0
* 1 (one-based indexing): The second element of the array is indexed by subscript of 1
* n (n-based indexing): The base index of an array can be freely chosen

**Advantages of using arrays:**

* Arrays allow random access of elements
* Arrays have better [cache locality](https://en.wikipedia.org/wiki/Locality_of_reference) that can make a pretty big difference in performance.

**Disadvantages of using arrays:**

* can’t change the size --because of static memory allocated

**Operations on Array :**

* 1. **array(data type, value list) -  create**an array
  2. **append()**:- **add the value**  at the end
  3. **insert(i,x)** :- **add the value(x) at the ith position**
  4. **pop()**:- This function**removes the element at the position** mentioned in its argument and returns it.
  5. **remove()**:- This function is used to**remove the first occurrence** of the value mentioned in its arguments.