# STL notes

* STD::ARRAY
  + Syntax: std::array<T, N> array;

// where T is data type and N is size of array

In array the access time is of the order 1 i.e, O(1).because we can access any data inside the array directly.

* + std::array is a container that encapsulates fixed size arrays.
  + array size is needed at compile time.
  + Assign by value is actually by value.
  + Access Elements:
    - a. at() //this throws out of bound exception
    - . [] //this gives a garbage value instead
    - c. front() //gives first element
    - d. back() //gives last element
    - e. data() // gives access to the underlying array i.e, its pointer
* STD::VECTOR
  + Syntax: std::vector<T> vector;
  + Vectors are array(O1) + linked list(Dynamic size)
  + When we push\_back an element inside the vector when there is no room left in the array to store elements then the class will automatically create an array double the size of current array and copy the element into that array and then the array pointer starts pointing towards the new array. And this process continues.so in case of vectors the size and capacity of the array will differ. Size()is no of live elements inside the array while capacity() can be more since it’s a dynamic array.but this makes the process slow so we reserve() some capacity at the start so that the creating and copying process doesn’t happen much thus making the program faster
  + std::vector is a sequence container and also known as Dynamic Array or Array List.
  + Its size can grow and shrink dynamically, and no need to provide size at compile time.
  + ELEMENT ACCESS: at(), [], front(), back(), data()
  + MODIFIERS: insert(), emplace(), push\_back(), emplace\_back(), pop\_back(), resize(), swap(), erase(), clear()
  + Other functions:reserve(num), size(), capacity()
  + syntax for 2D array

vector<vector<Type>> name(row,vector<Type>(col);

for creating a 2D vector with row rows and col columns.