Templates

Class → Object

Template → Class

↓

(Parameterized classes)

Why use templates?

→ DRY

→ Generic Progamming

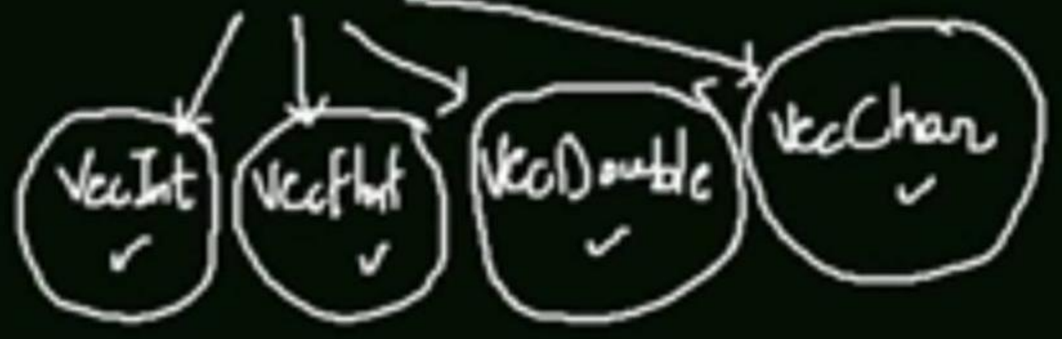
Class Vector {

int * arr's

int size;

Public:

DRY



Subscribe

Introduction to STL

→ Competitive Programming :

STL → Standard template library

Library of what ?

↓

→ Generic Classes and functions.

Limited time = → Resizize.
→ Sort
→ Search.

History : [HP → Alex. → Merg]

Why use STL? → Reuse : Well tested Components
→ Time Savings!

Introduction to STL

Components of STL → ① Containers → [→ Stores data
→ Use template classes

② Algorithms → [Sorting
Searching
Use template functions

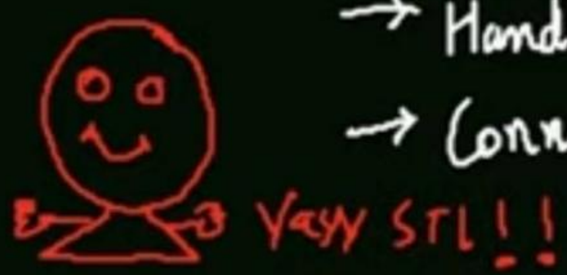
③ Iterators

Container:

← ↑ ← iterator moves as instructed by the Algo.

→ Object points to an element in a Container
→ Handled just like pointers
→ Connects Algo. with Containers.

STL is used because its a good idea not to Reinvent the wheel.



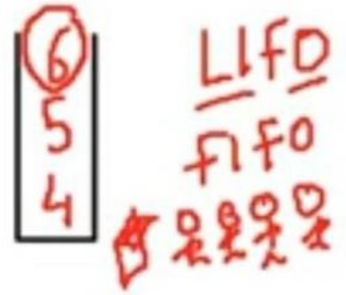
Containers in STL

STL = Containers + Algo + Iterators

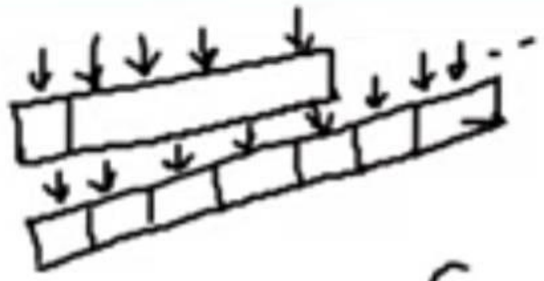
- Containers → object which stores data
- Algo → Procedure to pass data
- Iterators → Object which points to an element of a Container

Containers

1. → Sequence Containers → Linear fashion
   - Vector
   - List
   - Dequeue
2. → Associative Containers → Direct Access
   - Tree
     - → Set / Multiset
     - → Map / Multimap
3. → Derived Containers → Real world Modelling
   - → Stack ✓
   - → queue ✓
   - → priority_queue



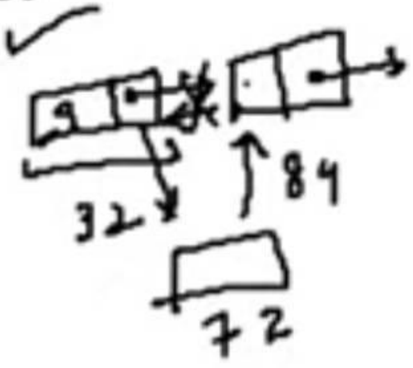
When to use which?



Sequence Containers

1. Vector → RA = Fast
   - M. Insertion / Del = Slow
   - Del / Insertion at the end = fast

2. List → RA → Slow
   - M Insertion → Fast
   - Del / Ins at the end → Fast



91

Data Structure

Associative Containers → All operations fast except RA ✓

Reshmi
Rohan
Rohit

Derived Containers → Depends → Data Structure