

Stacks

Stacks are declared as an ``abstract" class. They are currently implemented in any of three ways.

VStack

implement fixed sized stacks via arrays.

XPStack implement dynamically-sized stacks via XPlexes.

SLStack implement dynamically-size stacks via linked lists.

All possess the same capabilities. They differ only in constructors. VStack constructors require a fixed maximum capacity argument. XPStack constructors optionally take a chunk size argument. SLStack constructors take no argument.

Assume the declaration of a base element \mathbf{x} .

Stack s; or **Stack s(int capacity)** declares a Stack.

s.empty() returns true if stack s is empty.

s.full() returns true if stack s is full. XPStacks and SLStacks never become full.

s.length() returns the current number of elements in the stack.

s.push(x) pushes x on stack s.

x = s.pop() pops and returns the top of stack

s.top() returns a reference to the top of stack.

s.del_top() pops, but does not return the top of stack. When large items are held on

the stack it is often a good idea to use **top()** to inspect and use the top of stack, followed by a **del_top()**

s.clear()

removes all elements from the stack.