

The Stack: A Useful ADT

- an ADT that holds a collection of items where the elements are always added to one end
- last item pushed onto the top of a stack is the first item to be removed
- last-in-first-out data structure
- can be implemented with either an array or linked list

Stack Operations

- push something on top
- pop something off the top
- look at stack's top value

Implementing a Stack

```
1  const int SIZE = 100;
2
3  class Stack {
4      public:
5          Stack() {m_top = 0;}
6          bool push(int val) {
7              if (m_top >= SIZE) return false; // overflow!
8              m_stack[m_top] = val;
9              m_top += 1;
10         }
11
12         int pop() {
13             if (m_top == 0) return -1; // underflow!
14             m_top -= 1;
15             return m_stack[m_top];
16         }
17
18     private:
19         int m_stack[SIZE];
20         int m_top;
21 };
22
23 int main() {
24     Stack is;
25     int a;
26
27     is.push(5);
28     is.push(10);
29     a = is.pop();
```

```

30     cout << a;
31     is.push(7);
32 }

```

- stacks are available in the C++ STL
 - syntax is unorthodox
 - `std::stack<type> variableName;`
- has no maximum memory (only restricted by machine)

NOTE: NEVER USE “using namespace ____” IN A HEADER FILE!!

```

1 #include <stack>
2 using namespace std;
3
4 int main() {
5     stack<int> istack;    // stack of ints
6
7     istack.push(10);     // add item to top
8     istack.push(20);
9
10    cout << istack.top();  // get top value
11    istack.pop();         // kill top value
12
13    if (!istack.empty())
14        cout << istack.size();
15 }

```

Common Uses for Stacks

- storing undo items for your word processor
- evaluating mathematical expressions
- solve mazes
 - uses **depth**-first search
 - not as efficient as possible

All CPU's have stacks built in

- when you pass a value to a function, the CPU **pushes** that value onto a stack on the memory
- when your function returns, the values are **popped** off the stack and go away
- every time you declare a local variable, the program pushes it on the PC's stack automatically

Postfix Evaluation Algorithm

1. Start with the left-most token.
2. If the token is a number,
 - a. push it onto the stack.
3. Else, if the token is an operator,
 - a. pop the **top value into a variable called v2**, and the **second-top value into v1**.
 - b. Apply operator to v1 and v2.

4. If there are more tokens, advance to next token and go back to step #2
5. After all tokens have been processed, the top # of the stack is the answer

7 6 * 5 + —————> 42

The Queue: Another ADT

- like a line
- FIFO data structure
 - first in first out

Interface

- enqueue(int a);
 - insert an item on the rear
- int dequeue();
 - removes and returns top item from front of the queue
- bool isEmpty();
 - determines if queue is empty
- int size();
 - determines # of items in queue
- int getFront();
 - gives value on top without dequeuing

Common Uses

- data download from the Internet
 - download speed is faster than computer can display it to the user
 - “buffer”
- solving a maze
 - **breadth**-first search