
EECS 280 - DISCUSSION # 13

Week of April 7



Outline

- * Administrivia
- * Project 5
- * Functors
- * A Tree Grows...
- * Discussion Evaluations

Administrivia

- * Project 5

 - * Due April 15, 11:59 pm

- * Final Exam

 - * April 22, 10:30 - 12:30

 - * Sample on CTools

- * Next Week: Optional Review

Outline

- * Administrivia
- * **Project 5**
 - * Clarification
 - * Stacks & Queues
- * Functors
- * A Tree Grows...
- * Discussion Evaluations

Clarification

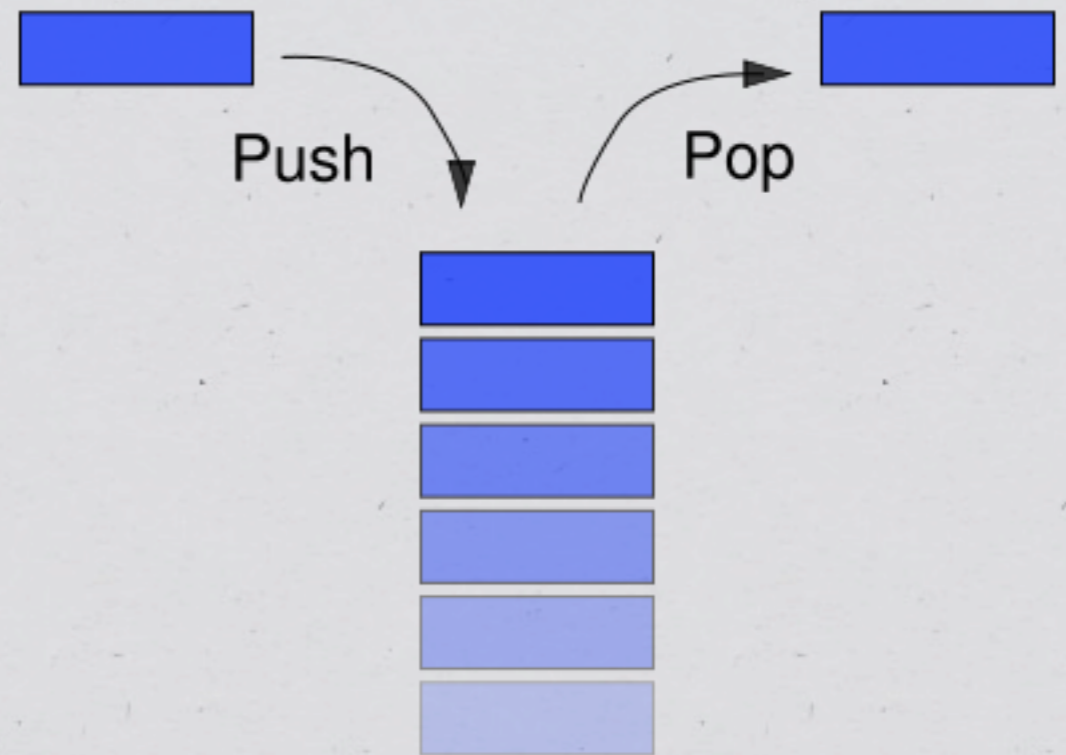
```
int *x = new int(5);  
list.insert( x );
```

```
*x = 7;  
delete x;
```

The list owns the pointer address, not the data stored at that address.

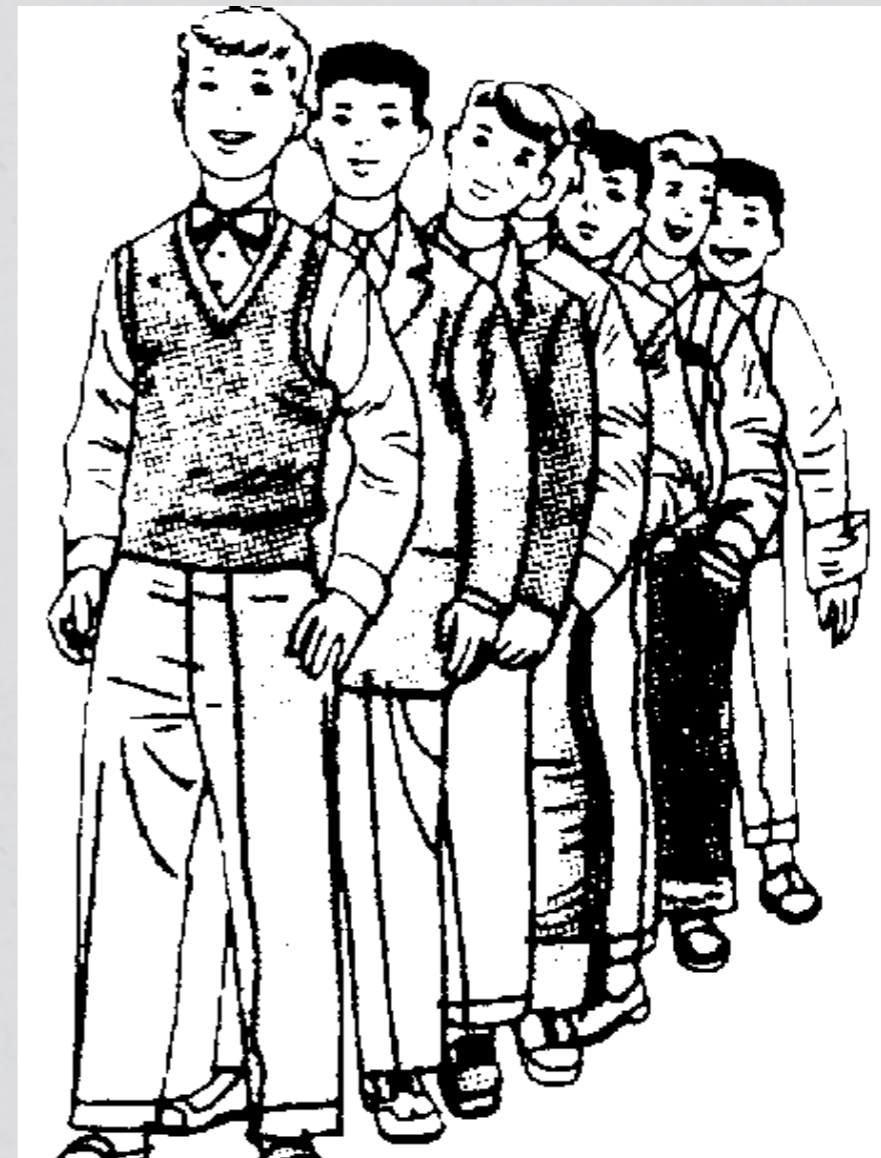
Stacks

- * Stacks are abstract data types based upon LIFO (last in, first out)
- * Basic operations
 - * Push
 - * Pop
 - * Peek
 - * Length

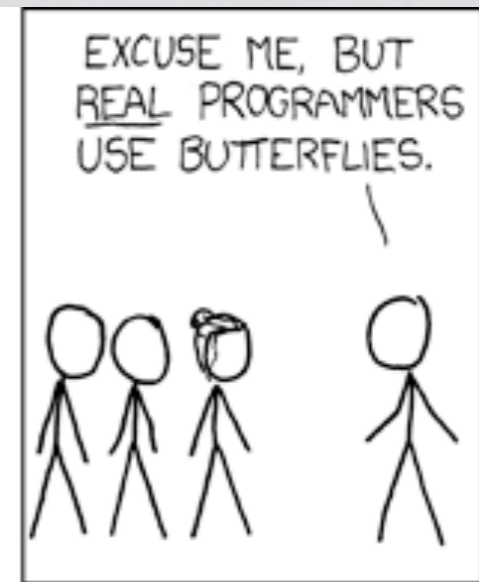
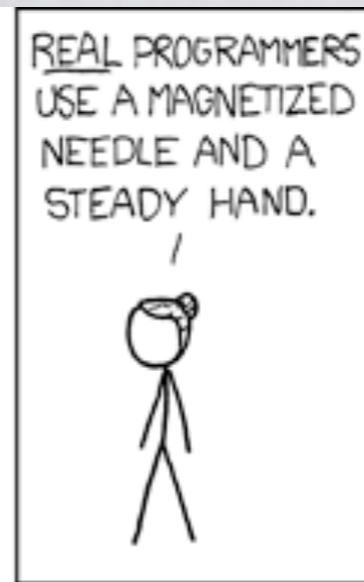
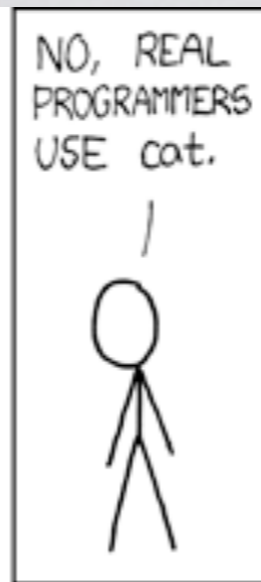
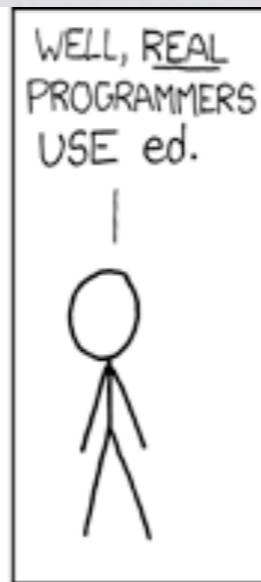
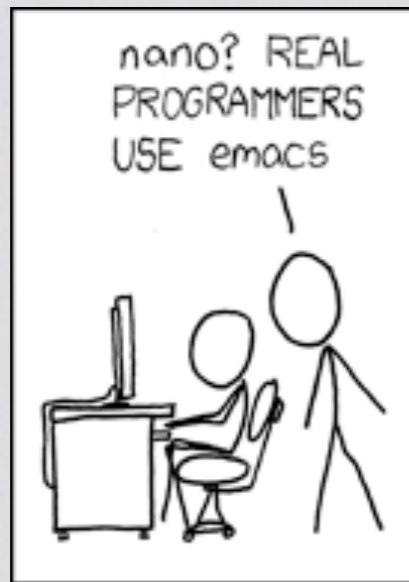


Queues

- * Queues are abstract data types based upon FIFO (first in, first out)
- * Basic operations
 - * Push
 - * Pop
 - * Peek
 - * Length



Interlude

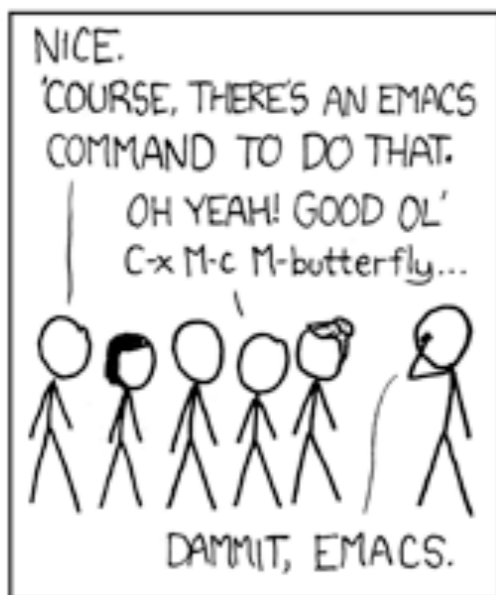
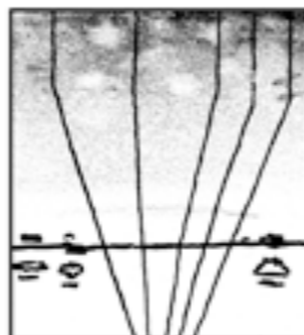


THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.



THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,

WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.



Outline

- * Administrivia
- * Project 5
- * **Functors**
- * A Tree Grows...
- * Discussion Evaluations

Remember Function Pointers

```
int count_predicate( list_t list, bool (*fn)(int) )
// EFFECTS: returns the number of elements in list
//           for which fn() returns true
{
    int counter = 0;
    while ( !list_isEmpty( list ) )
    {
        if ( fn( list_first( list ) ) )
            counter++;
        list = list_rest( list );
    };

    return counter;
}
```

What happens if we need something more flexible than a function alone to serve our purposes?

What about saving state?

Functors

- * A functor, or function object, is an object designed to work as a function
- * In C++ this is achieved by overloading the parenthesis operator
- * A functor provides more flexibility than a function pointer via member variables, constructors, and other functions

Functor Example

```
class Predicate {
public:
    virtual bool operator()(int n) = 0;
}

class Counter : public Predicate {
    int count;
public:
    Counter() { count = 0; }
    getCount() { return count; }
    bool operator()(int n) { if (n==5) count++; return false; }
}
```

Functor Usage

```
bool search_list( list_t &list, Predicate &found )
{
    while ( !list_isEmpty(list) && !found( list_first(list) ) )
        list = list_rest( list );

    return !list_isEmpty( list );
}

int count_predicate( list_t &list )
{
    Counter pred;
    search_list( list, pred );
    return pred.getCount();
}
```

Outline

- * Administrivia
- * Project 5
- * Functors
- * **A Tree Grows...**
- * Discussion Evaluations