CSE 101-01 2-14-23

- Pay: due tonite
- ADTs in C++
- Pa 5

struct in C:

```
struct Blah {
    int foo;
    int bar;
};
struct Blah x;
x.foo = 6;
x.bar = 7;
```

struct in C++

```
struct Blah {
    int foo;
    int bar;
    Blah(int a, int b){
        foo=a;
        bar=b;
    }
};
:
Blah x(6,7);
```

class in C++

```
class Blah{
    int foo;
    int bar;
    Blah(int a; int b){
        foo=a;
        bar=bj
    }
    void set_foo(int a){
        foo =a;
    }
};
.
Blah x(6,7);
x.set_foo(8);
```

basic skeleton:

```
class Blah{

Private:
    // inner classes or structs
    // variables
    // Private functions

Public:
    // constructor / copy constructor
    // deconstructor
    // member methods i.e. ADT OPs
    // overloaded OPs

};
```

To build an ADT

Files: Blah.h, Blah.cpp

Blah.h:

```
class Blah {

Private:
    // inner struct
    // member vars & helper fcns.

Public:
    // Prototypes of ADT ops.
    // also over-loaded OPS.

};
```

Blah.cpp

scope resolution operator ::

```
// an ADT op. defn
type Blah::fcn1(.....){
    :
}
type Blah::fcn2(....){
    :
}
// private helper fcn, member
type Blah::helper(...){...}
// a non-member helper fcn
type helper2(....){....}
```

Ex. Queue ADT in C++

The this pointer

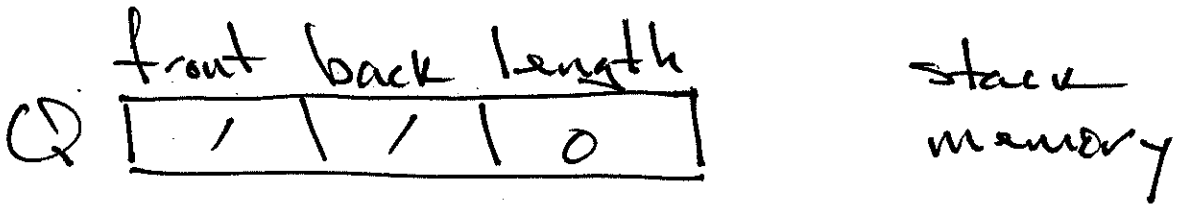
every member fcn has access to this.

when an op (say fcn()) is called, we do

A.fcn()

instance of Queue } the implicit 1st argument to fcn()

Empty state:



non-empty state: