Memory Management

Kate Gregory

@gregcons www.gregcons.com/kateblog

The Free Store

Local variables go out of scope when the function ends

That's not always what you want

The free store is for longer lived variables

The Free Store

Create with new

Returns a pointer to the object or instance

Uses a constructor to initialize the object

The Free Store

Tear down with delete

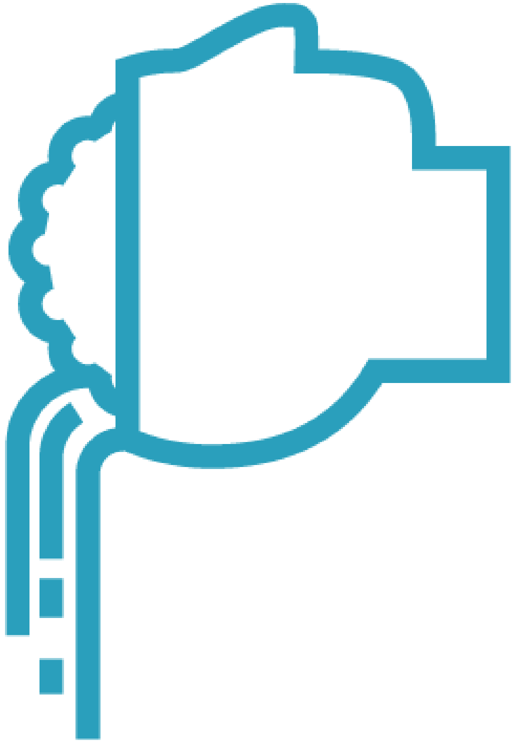
Uses the destructor to clean up the object

The Free Store

Slightly different syntax for “raw arrays”

But modern C++ avoids “raw arrays”

Manual Memory Management



If you got a pointer, from `new`, you have to keep track of it

- At some point you must call `delete`

What happens if someone copies it?

What happens if the local variable (the pointer) goes out of scope early?

Manual memory management is hard, with a variety of mistakes to make

- Delete too soon
- Delete twice
- Never delete

Rule of Three

Destructor

- Deletes what may have been created with new

Copy constructor

- Uses new to initialize from existing value

Copy assignment operator

- Deletes, then uses new to initialize

Became Rule of Five

Move constructor

Move assignment operator

Best: Rule of Zero

Design your class not to need any of these

- Written by you, anyway

Stack semantics

Easy Memory Management

C++11 has a nice range of smart pointers

- They do all this for you

Imagine a template class with just one member variable

- A T* that you got from new

Constructor saves the T* in the member variable

Destructor will delete that T*

- No memory leak

Handle copy one of two ways

- Prevent it (private copy constructor and copy assignment operator)
- Have a reference count: copy increments, destructor decrements, delete at 0

The key thing: operator overloads

- *
- ->

Standard Library Smart Pointers

shared_ptr

- Reference counted

weak_ptr

- Lets you “peek” at a shared_ptr without bumping the reference count

unique_ptr

- Noncopyable (use std::move)

Summary

The free store (aka the heap) gives objects a lifetime longer than local scope

Manual memory management is hard

Smart pointers make life a lot simpler