

Where to from here?

Objectives

- To provide a starting point for any person who wants to continue learning about C++ programming.
- To provide examples of code that use these starting points, to provide a base for bigger things.

What we've covered so far

- C++ is a big language - too big to cover in a single semester.
- We've looked at
 - OO design methodology and OO through classes
 - Generic programming through templates
 - Optimization
 - The Standard Template Library
 - Exception handling

Surely that's enough?

- C++ supports many different paradigms
- C++ has a lot of hidden catches that trap the unwary programmer.
- Proper use of exceptions could be a subject in itself!
- The STL has many algorithms and containers we haven't used

What now?

- Start by reading books that will make you a better C++ programmer.
- Practice what you find in those books in your applications.
- Write code. Practice makes perfect.
- Be critical with your code - acknowledge bad code, and fix it.
 - Never be afraid to fix something that is bad!

A few books

- *Effective C++* and *More Effective C++*, by Scott Meyers.
 - These are a must read.
 - The first book contains 50 hints to improve your code
 - The second book contains 35 more hints.
 - Also available as a combination of both books in CD form
 - I prefer the 'dead-tree' editions as it's hard to read a CD in bed!

A few books

- *Exceptional C++*, *More Exceptional C++*, and *Exceptional C++ Style*, by Herb Sutter
 - A set of problems of varying degrees of difficulty.
 - Later books reinforce earlier books, but add a lot more information as well
 - Makes you realise there are hidden implications of certain actions.
 - Highly recommended.

A few books

- For optimization, we used *Efficient C++* by Bulka and Mayhew as the reference.
- Filled with 'war stories' from their time at IBM
- Fascinatingly detailed look at computer architecture and its relationship to optimization, but this chapter is not for the faint-hearted.
- Lots of spelling mistakes throughout the book 😞

A few books

- To succeed at C++, you will need an excellent reference book.
 - Everyone has their own favourite. Mine is Stroustrup.
 - "*The C++ Programming Language, 3rd Edition*" by Stroustrup. (Special Edition is fine)
 - Contains everything in the language you need to know
 - Excellent summary of the STL functions.
 - Sometimes need to decipher what he means (although he is consistent)

Magazines

- Books are very static by nature - they rarely get updated.
- Magazines provide up-to-the-minute details on C++, and the latest crazes.
- Often **much** cheaper to subscribe than to buy in Australia.
 - (e.g. US\$85 2-year subscription to (\$16 an issue) Circuit Cellar, with \$1 = US\$0.77, equals \$4 an issue)

A few magazines

- The C/C++ Users Journal
 - A bit thin, but lots of details within articles.
 - At \$16 an issue, worth buying when an article grabs your fancy
 - Herb Sutter writes a "guru" column, well worth a read.

A few magazines

- Dr Dobbs Journal
 - Not C++ specific, and losing focus a bit in recent years
 - Worth a look every so often.
 - Worth conning your boss into getting a subscription for you!

A few magazines

- Taking a look in a **good** newsagent will probably reveal a few more.
- Again, just buy when an article takes your fancy.

The internet

- “The internet is so big and amazing that for some people it is a complete substitute for life”

Andrew Brown.

- There are plenty of websites, newsgroups, etc. to keep up with or provide information.
- Read comp.lang.c++ if you are into long Usenet flamewars (“Y-M NOW” anyone?)

Does Google know?

- Search engines (Google, Bing, Yahoo) are very powerful tools to solve your programming questions!
- Don't hesitate to search in the Web, there is a high probability someone has had the same question before you!
- Very convenient when stuck during programming.

Graphics and GUIs

- Plenty of windowing toolkits out there. I've had experience with Qt and (long ago) wxWindows.
- I liked Qt and disliked wx, but both have undergone radical changes. Wx is supposed to be good now!
- How about Eclipse?
- <http://www.trolltech.com/qt/>
- <http://www.wxwindows.org/>
- <http://www.eclipse.org>

Graphics and GUIs

- Qt provides a lot of functionality and **widgets** (GUI functions, like a check box, or list).
- It even provides multithreading and some networking support.
- Qt forms the basis for the KDE desktop in Linux.
- There are also gtk (the foundation of the Gnome desktop)

Graphics and GUIs

- For 3D, OpenGL is a C library, but works well with C++.
- Nigel Stewart has written a C++ wrapper for OpenGL. <http://www.nigels.com/> is likely to have it.
- DirectX is Windows specific. Avoid it.
- Interactive 3D and Real-time Rendering are two courses offered by RMITCS.

Windows programming

- /me shudders
- Steer clear of MFC and stick to Win32 API programming. Unfortunately(?) this is mainly C.
- MSDN has the best documentation (and also the most conflicting!)
- Stick to a windowing toolkit like Qt or wxWindows.

Multithreading

- Check out ptypes. It doesn't play nicely with the standard C++ string, so be prepared to do some namespace scope resolution (and not use `using std::string`)
- It has thread classes that wrap the native threads.
- Do Operating Systems for a (brief) introduction.

Multithreading

- In `ptypes`, you derive from `pt::thread`, and override the `execute()` member function.
- You do the same in Qt, but with a `QThread` object, and the `run()` member function.
- Choose one or the other, but not both! Providing you don't inherit from both `QObject` and `pt::thread` you should be OK.

Multithreading Gotcha

- One common arrangement is to use a member thread to call a member function within an object.
- Don't attempt to start the thread in the constructor of the object.
- Chances are it will be scheduled and attempt to call the member function before the constructor finishes.

Networking

- Again, ptypes is worth a look.
- It works similar to the Java API's Socket and ServerSocket classes.
- Simple code can be produced in under an hour to talk to another host.

Networking

```
#include <pinet.h>
#include <iostream>

const pt::ipaddress(123,45,67,89);

int main() {
    pt::ipstream stream(IP, 110);
    stream.write("USER anryan\r\n");
    stream.write("PASS 12345678\r\n");
    stream.write("STAT");
    pt::string input = stream.line();
    std::cout << "You have " << cnt << " messages";
    stream.write("QUIT");
    stream.close();
    return 0;
}
```


Boost library

- Boost is a library of various classes that may in the future find their way into the C++ standard.
- Many helpful classes
 - Parsers, mathematical functions, etc.
- Free to use.
- <http://www.boost.org/>

Links

Links:

Qt: <http://www.trolltech.com/qt/>

wxWindows: <http://www.wxwindows.org/>

ptypes: <http://ptypes.sourceforge.net/>

OpenGL: <http://www.opengl.org/>

Doxygen: <http://www.doxygen.org/>

XP: <http://www.extremeprogramming.org/>

Done!

- You've made it!
- Feel free to put me in for a teaching award 😊