COMP6771
Advanced C++ Programming

Week 1.1
Course Outline

Teaching Staff

Lecturer in charge
Imran Razzak

Course Admin
Simon Haddad

Potential
Guest Lecturers

Christopher Di Bella
Nathaniel Shead
Matthew Stark
Optiver

Tutors

Giuseppe Redondi
Kaiqi Liang
Max Owen
Ryan King
Shane Kadish
Simon Haddad

Course Objectives

You will develop:

1. skills in writing software using C++20
2. skills in using libraries to develop software
3. skills in using tools to build and test software
4. knowledge and understanding about unit testing
5. knowledge and understanding about reactive programming, object-oriented programming, and generic programming

What is C++?

- Lightweight-abstraction programming language
- Lets you use the right abstractions at the right time
- C++ is a more complex language than C, but C++ code is much simpler than C.
- Like C, C++ is written for running on hardware directly
- Has OOP capabilities, but not required (unlike Java)

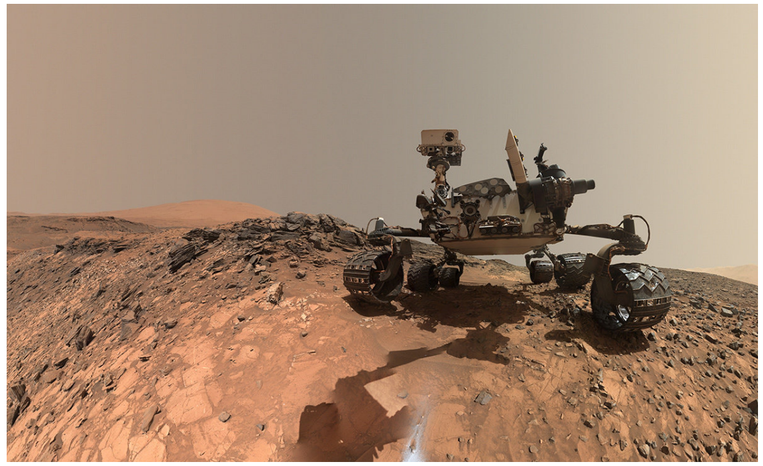
C++ Design Pillars

- Don't leave room for a language between C++ and assembly.
- Abstractions should have as little cost as possible.

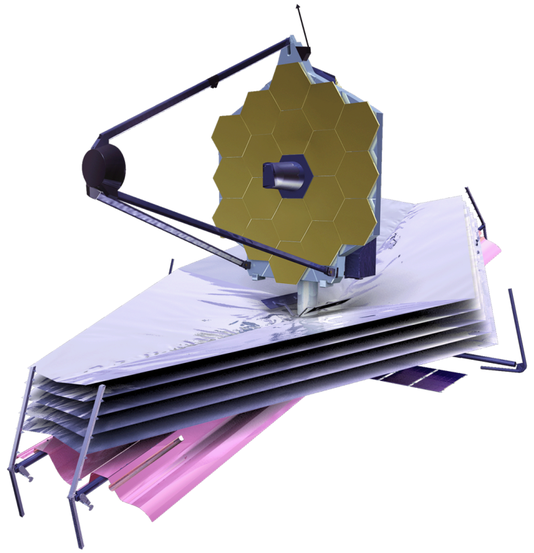
C++ is not C

- C++ is backwards compatible with C, so it's easy to think that you can build your C++ understanding directly on top of your C understanding
- However, while valid C code is often valid C++, good C is is almost never good C++ code. Over the years C++ continues to diverge from C
- For example, when we teach you best practice, we will not be using:
  - malloc
  - free
  - C-style arrays
  - C-style strings
- And will be sometimes discouraging use of:
  - raw pointers (char *, int *)

What's C++ good for?



Mars Curiosity Rover, courtesy NASA/JPL-Caltech.

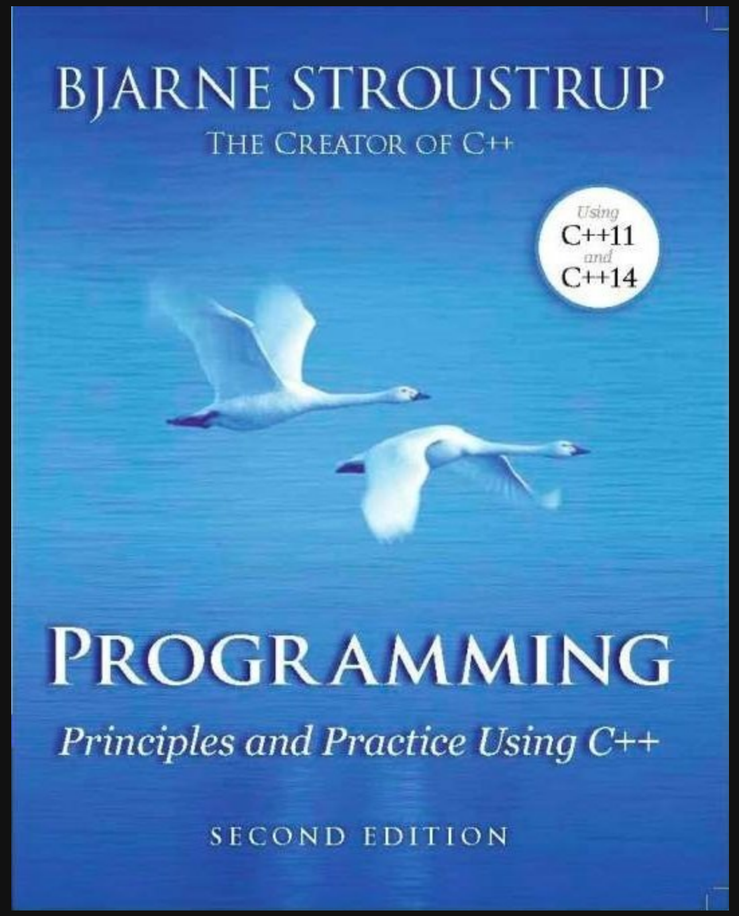


James-Webb Telescope, courtesy NASA/JPL-Caltech.

What is C++ used for?

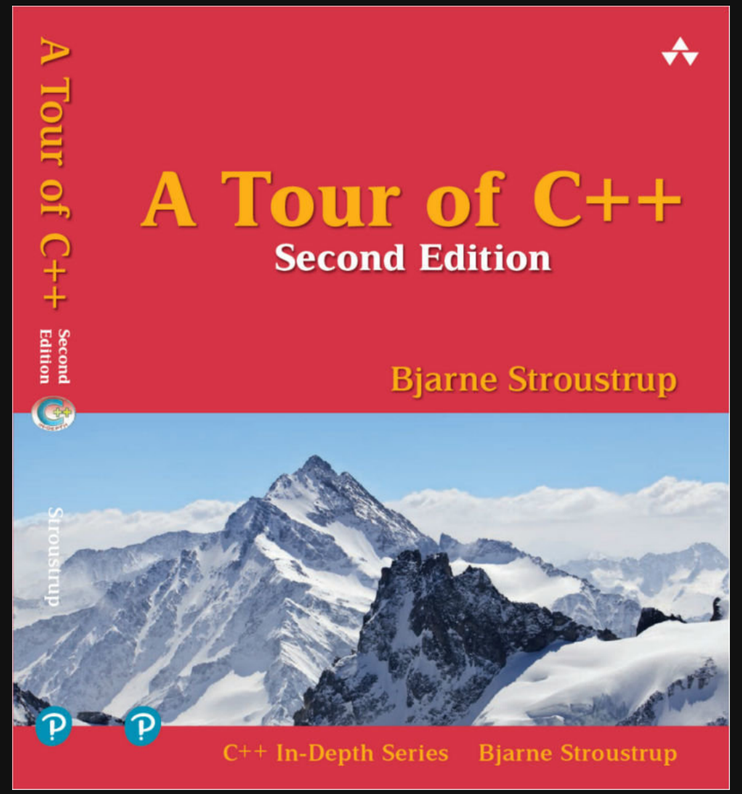
- Operating systems
  - Despite myths, C++ is as fast as C, but far more concise and easier to write correct code
- Low latency software (eg. high frequency trading)
  - C++ can have direct control over the hardware
- Games
- Just about anything you can think of

Learning Resources



Comprehensive intro to C++ (>100 pages of exercises!)

+



Will help you in a pinch (e.g. before exams and interviews)
Also covers newer stuff the Swan book doesn't

Learning Resources

cppreference.com

Good for looking up APIs and recalling language rules

DO NOT USE CPLUSPLUS.COM

code.visualstudio.com

Documentation on how to use the course editor

Where to get help

Your question/answer hierarchy:

1. Edstem forum
2. Your tutor (see Timetable page for links)
3. Lecturers (cs6771@cse.unsw.edu.au)
4. Imran (imran.razzak@unsw.edu.au)

Questions that are non-sensitive will only be answered on the forum

Schedule & Structure

- See course outline for full course schedule
- Weekly teaching provided includes:
  - 4 hours of lectures
  - 1 hour of tutorial
  - 7+ hours of recommended practice and associated work

We may provide additional material and webinars to assist in your learning. While these will be recommended, they will not be required.

Assessment

Assessment	Weighting	Due Date
Assignment 1	15%	Late Week 3
Assignment 2	25%	Early Week 7
Assignment 3	30%	Early Week 10
Exam	30%	Exam Period

Assignment due dates are subject to change (never earlier), so always see the assignment specification for more information

Assessment

- Final exam may be scaled
- Final exam:
  - No hurdle
- Assignments:
  - have an emphasis on testing
  - rely on version control (assumed knowledge)
  - have a late penalty outlined in the specification
- Plagiarism will not be tolerated.
  - Immediate zero for assignment.

Gitlab

This course is taught on gitlab.

For every tutorial (9) and every assignment (3) we will automatically deploy new repositories and subsequent changes to those repositories in your gitlab account.

If you are not familiar with git, or haven't used vlab in course before, we encourage you to check out git101 on the tutorials page.

If you're really out of your depth, you can always post on the forum. Your tutor will demonstrate a bit more of this in week 1.

We will discuss Gitlab more in the next lecture

Feedback

#include <C++>

Feedback