COMP6771
Advanced C++ Programming

Week 1.1
Course Outline

Teaching Staff

Lecturer in charge

Hayden Smith

Lecturers

Hayden Smith
Christopher Di Bella
Matthew Stark

Tutors

Adrian Martinez
Gary Bai
George Fidler
Jason Zavaglia
Josephine Anugerah
Mirette Saleh
Nathaniel Shead
Oliver Richards
Rahil Agrawal
Ryan Fallah
Saeed Baig
Simon Hadded

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++?

What is C++?

- Lightweight-abstraction programming language

What is C++?

- Lightweight-abstraction programming language
- Lets you use the right abstractions at the right time

C++ Design Pillars

C++ Design Pillars

- Don't leave room for a language between C++ and assembly.

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?

What's C++ good for?

What's C++ good for?

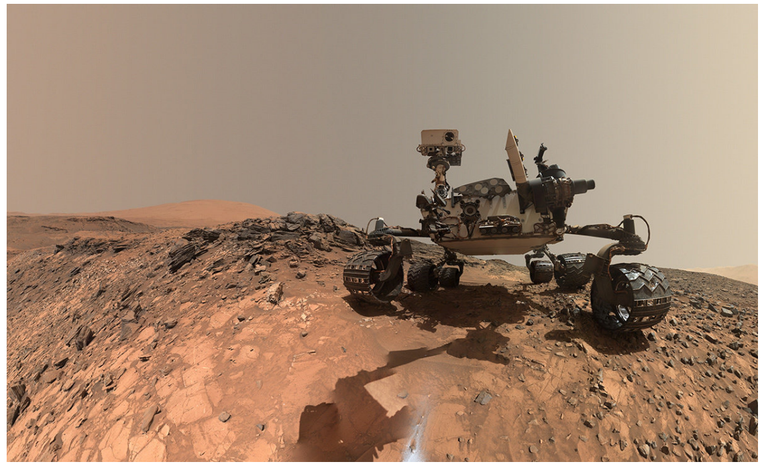
What's C++ good for?

Morgan Stanley

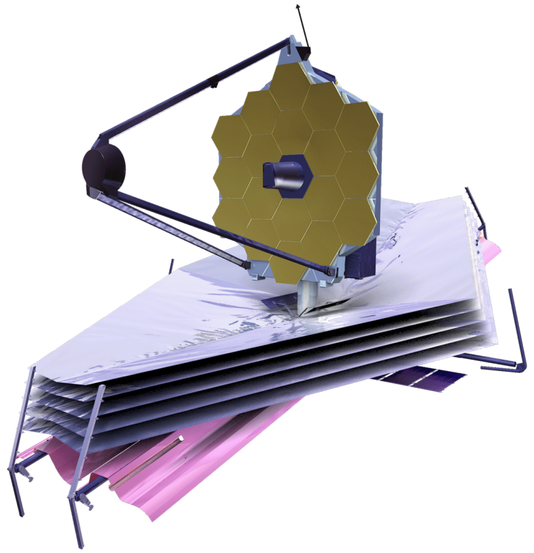
What's C++ good for?

Morgan Stanley

What's C++ good for?



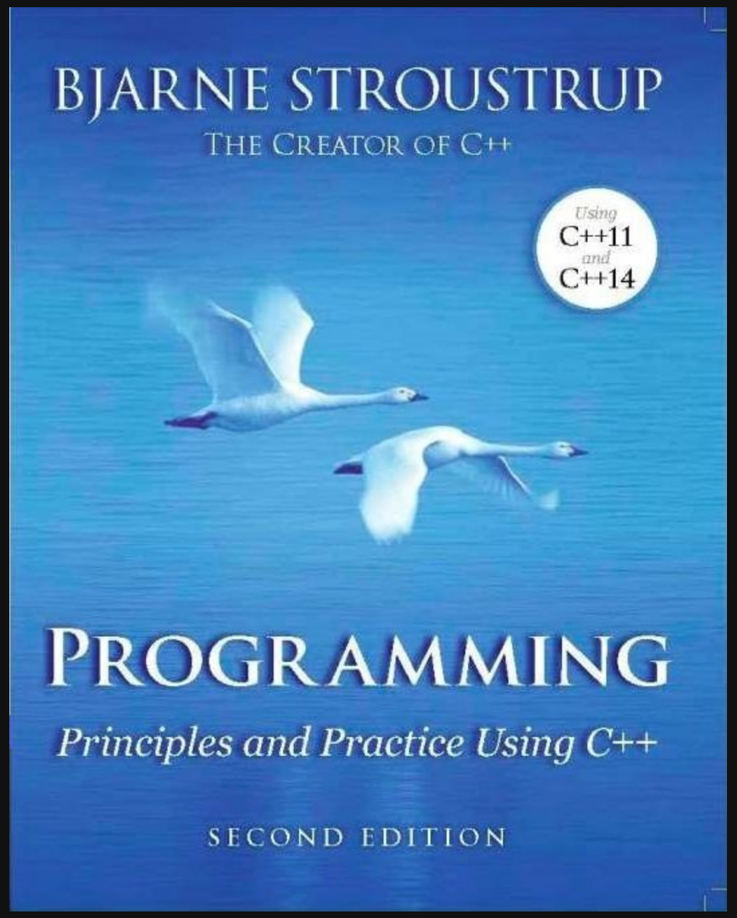
Mars Curiosity Rover, courtesy NASA/JPL-Caltech.



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

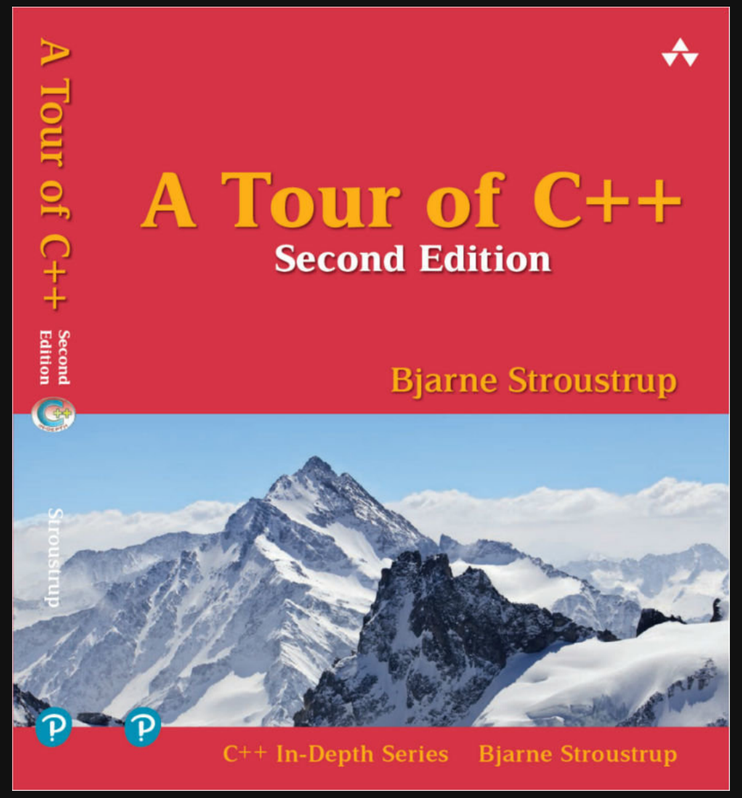
#include <C++>

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**
abseil.io	Good for looking up APIs and getting tips on how to use C++
code.visualstudio.com	Documentation on how to use the course editor

Where to get help

Your question/answer hierarchy:

1. Piazza forum
2. Your tutor (see Timetable page for links)
3. Lecturers (cs6771@cse.unsw.edu.au)
4. Hayden (hayden.smith@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	15%	Early Week 7
Assignment 3	20%	Early Week 10
Exam	50%	Exam Period

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

Assessment

Assessment

- Final exam may be scaled

Assessment

- Final exam may be scaled
- Final exam:
  - Hurdle exam

Assessment

- Final exam may be scaled
- Final exam:
  - Hurdle exam
- Assignments:
  - have an emphasis on testing
  - rely on version control (assumed knowledge)
  - have a late penalty outlined in the specification

Assessment

- Final exam may be scaled
- Final exam:
  - Hurdle exam
- 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.

E.G. For me personally: https://gitlab.cse.unsw.edu.au/z3418003/20t2-cs6771-tut01

Gitlab

If you are not familiar with git, or haven't used vlab in course before, we encourage you to check out the git instructions provided in the first tutorial.

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.

Feedback

During your first tutorial your tutor will ask for a few volunteers to be added to private channel with tutors to discuss and share thoughts around how they're finding the course.