

Overview of C++: Strategies for Learning C++

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

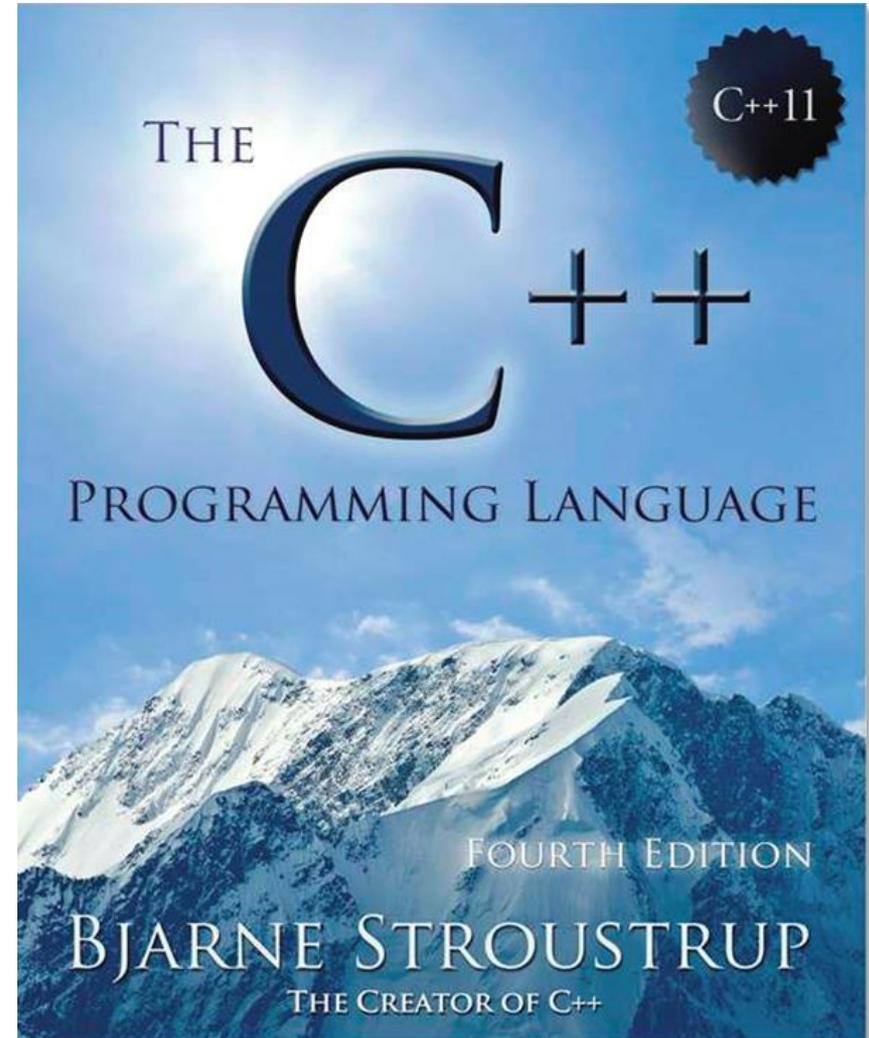
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Recognize the key components of C++
- Know strategies for learning C++



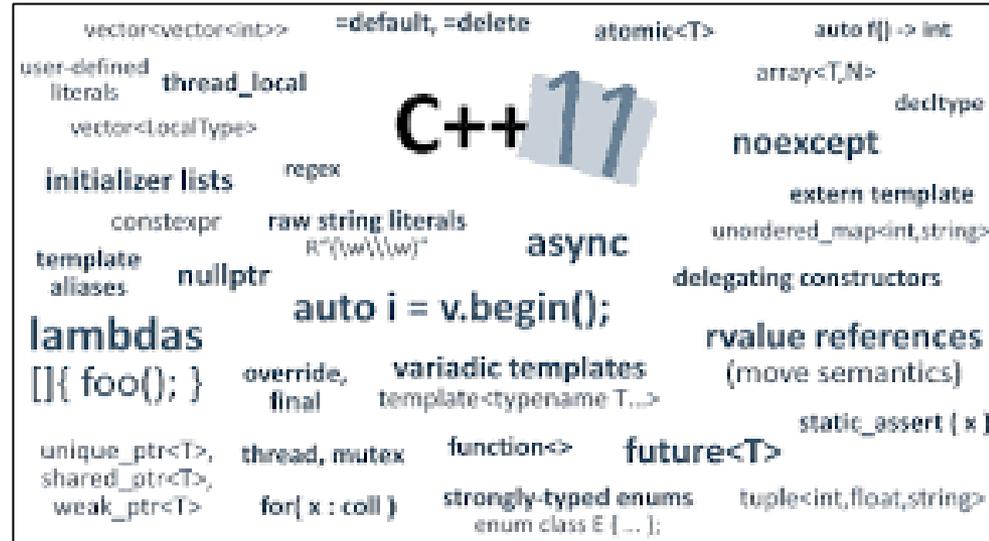
Strategies for Learning C++

Strategies for Learning C++

- C++ is a big language with a long history, so it's important to learn it smartly!

1. Focus on core concepts & design & programming techniques

- Don't get lost in every new-fangled C++ language feature



Many newer C++ features are intended for library developers, not app developers

Strategies for Learning C++

- C++ is a big language with a long history, so it's important to learn it smartly!
 1. Focus on core concepts & design & programming techniques
 2. Learn C++ to become a better software developer
 - i.e., become more effective at developing & testing modular & robust programs



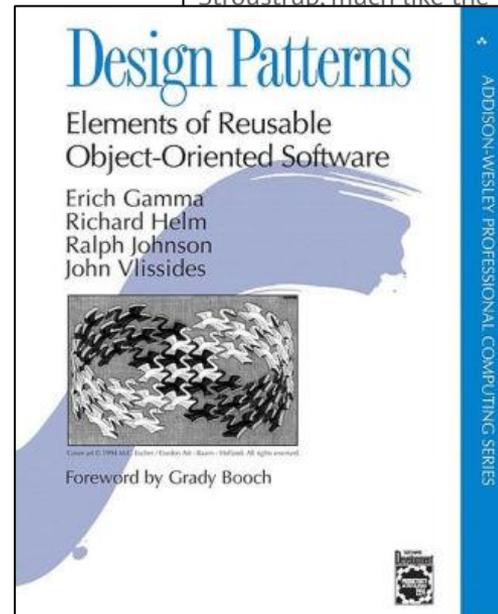
Strategies for Learning C++

- C++ is a big language with a long history, so it's important to learn it smartly!
 1. Focus on core concepts & design & programming techniques
 2. Learn C++ to become a better software developer
 3. Learn & apply software patterns & idioms
 - C++ supports many different programming styles

C++ Core Guidelines

“Within C++ is a smaller, simpler, safer language struggling to get out.” – *Bjarne Stroustrup*

The C++ Core Guidelines are a collaborative effort led by Bjarne Stroustrup, much like the C++ language itself. They are the result



discussion and design across a number of languages. The book encourages general applicability and that the code can be freely copied and modified to suit needs.

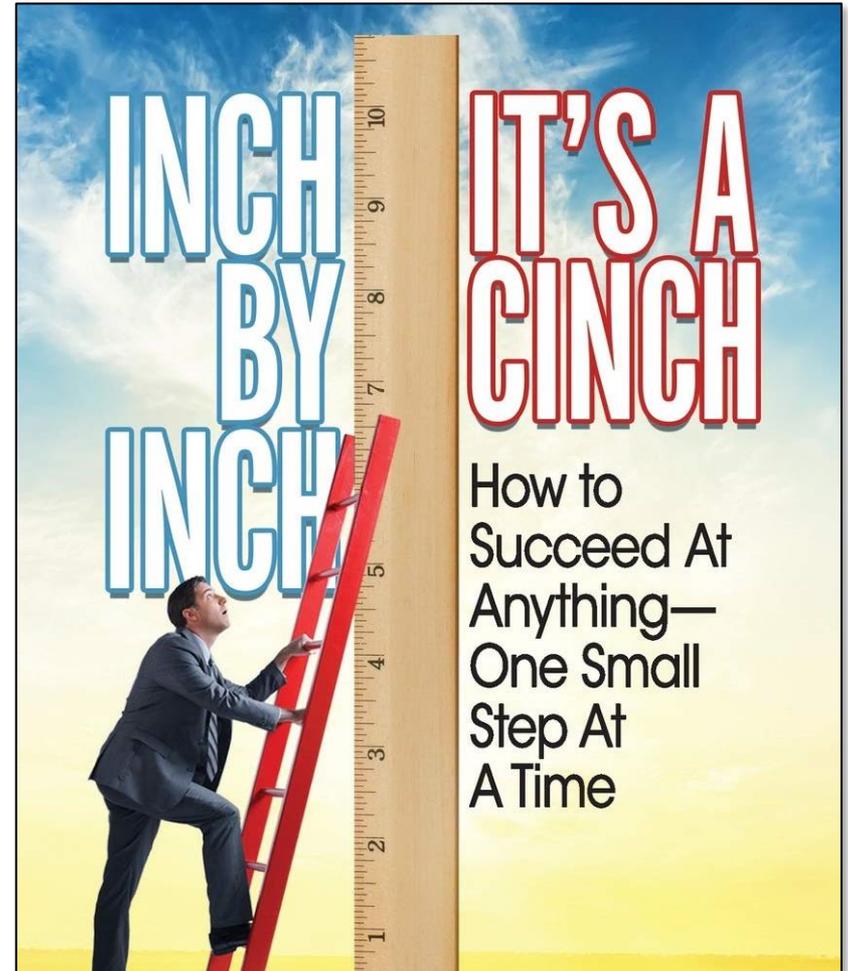
to help people to use modern C++ (we mean C++11 and C++14 (and soon C++17)). What would you like your code to look like in 10 years' time? You can start now?

on relatively higher-level issues, such as design, management, memory management, and object application architecture and library design. This will lead to code that is statically type safe and catches many more programming errors than is common in code today. And it will run fast -

you can afford to do things right.

Strategies for Learning C++

- C++ is a big language with a long history, so it's important to learn it smartly!
 1. Focus on core concepts & design & programming techniques
 2. Learn C++ to become a better software developer
 3. Learn & apply software patterns & idioms
 4. Learn C++ gradually
 - Don't have to know every detail of C++ to write good C++ programs



End of Overview of
Strategies for Learning C++
