

Software Development with C++

Mark Slater

Tom Latham

Luke Kreczko

Matt Williams


Ben Morgan



UNIVERSITY OF
BIRMINGHAM



University of
BRISTOL



Programming is in
top 5 most useful
skills gained by PhDs



**“48% were employed
in universities...”**

- STFC PhD Careers

Our Aim: "mpags-cipher"

Develop a command line application using the C++ programming language to encrypt/decrypt simple text messages using classical ciphers.

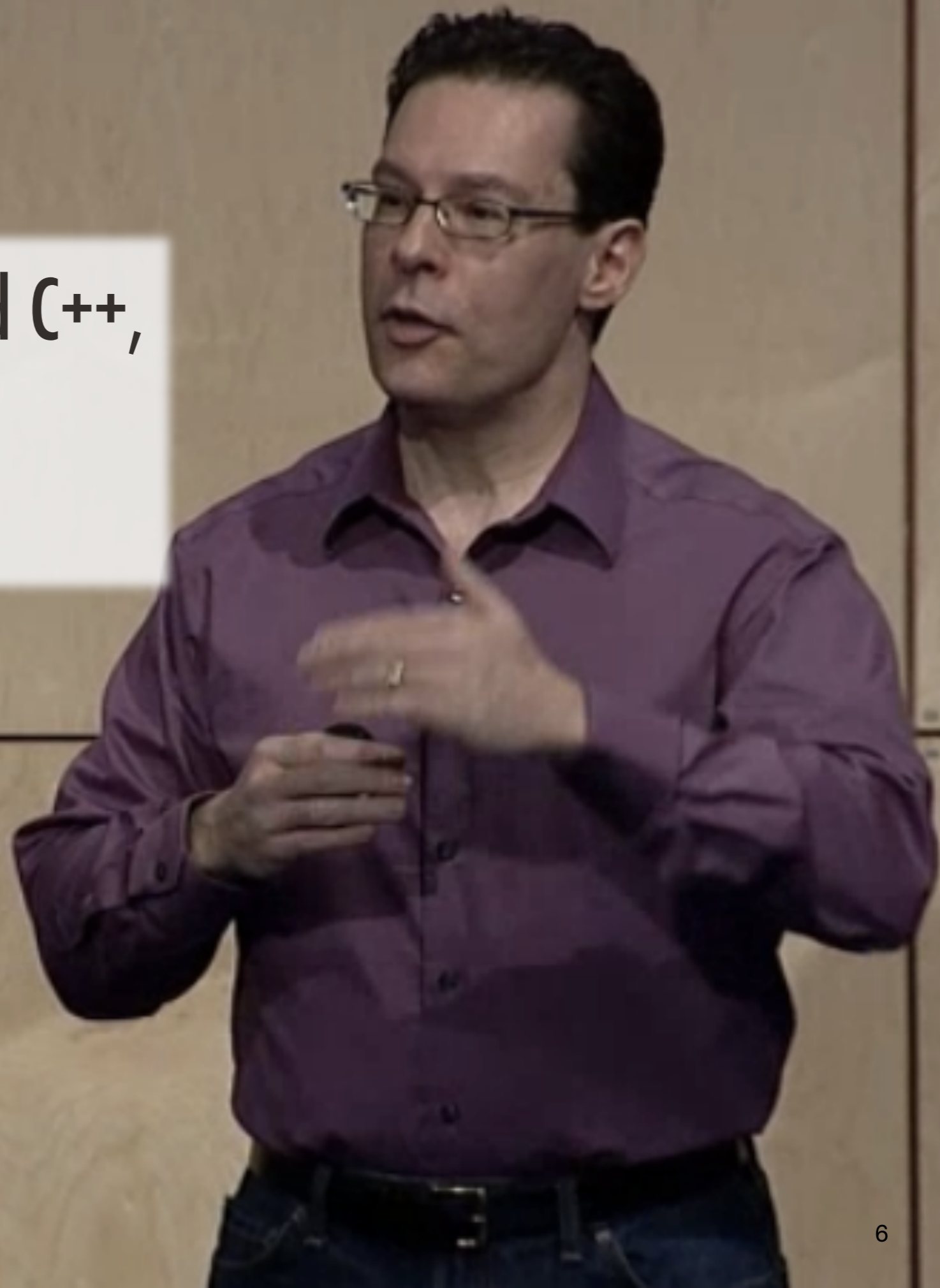
Requirements

- Read Plaintext from Keyboard or File
- User select of Cipher, Key and whether to Encrypt/Decrypt
- Provide Encryption/Decryption via
 - Caesar Cipher
 - Playfair Cipher
 - Vigenere Cipher
- Write cipher text to Screen or File



**“The world is built on C and C++,
did you know that?”**

- Herb Sutter



lambdas

```
[] {foo();}
```

constexpr

initializer lists

regex

C++11

nullptr

```
std::shared_ptr<T>,  
std::unique_ptr<T>,  
std::weak_ptr<T>
```

```
auto i = v.begin();  
for(x : collection)
```

“Not Your Parents’ C++” - Herb Sutter

lambdas

```
[ {foo();}
```

constexpr

initializer lists

regex

C++



98

nullptr

```
shared_ptr<T>,  
unique_ptr<T>,  
weak_ptr<T>
```

```
auto i = v.begin();  
for(x : collection)
```

“Your Parents’ C++” - Still much existing code

generic lambdas

```
[](auto x){foo(x);}
```

C++



binary literals

```
auto bin = 0b0100;
```

digit separators

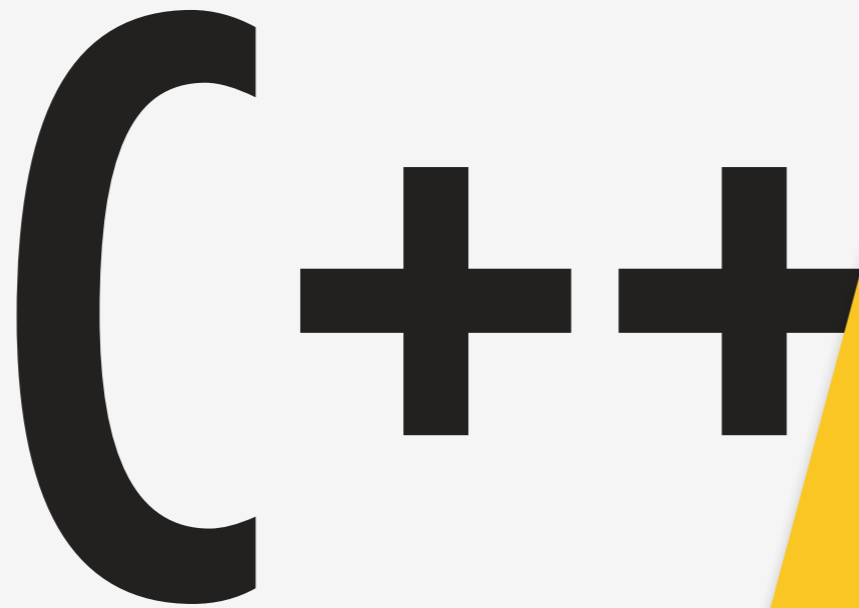
```
auto i = 1'000;
```

deduced return types

```
auto myFunc();
```

string views

`std::string_view` `constexpr` initializer lists



filesystem operations

`std::filesystem`

structured bindings

```
auto [ first, second ] = myPair;
```

parallel algorithms

mathematical

language constants

`std::numbers::pi` `constexpr` `initializer lists`

C

++



concepts

`std::movable`

`std::floating_point`

**associative
contains**

`myMap.contains(2);`

using enum

1. Procedural C++

2. Procedural C++ 2

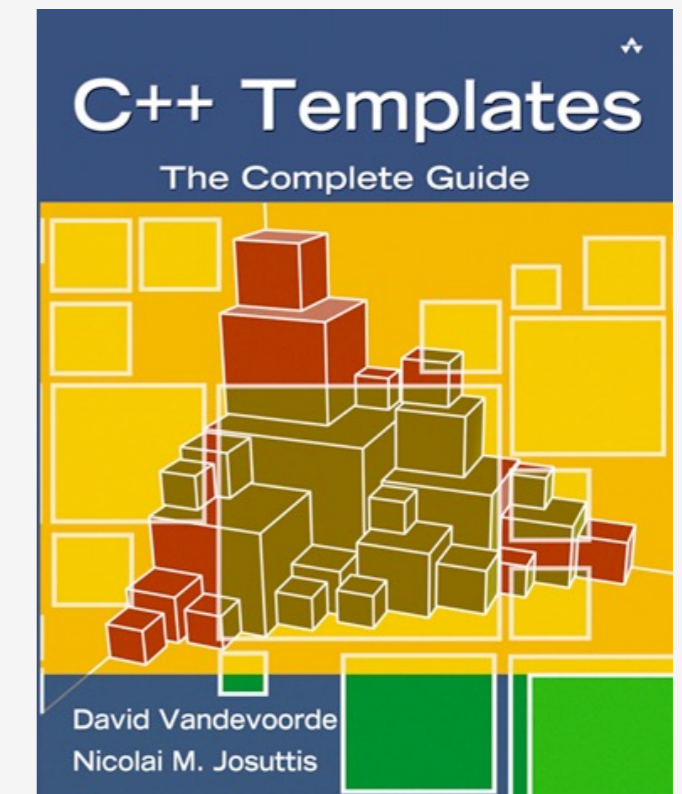
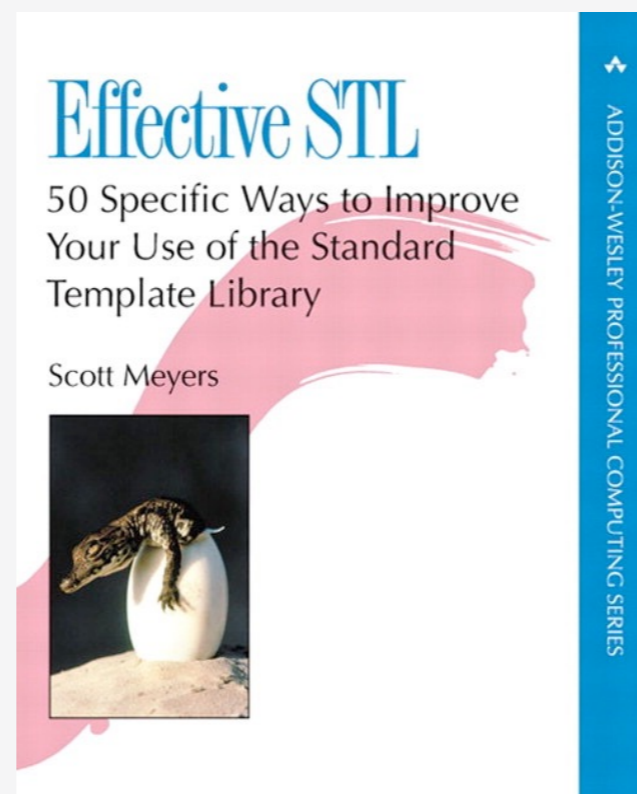
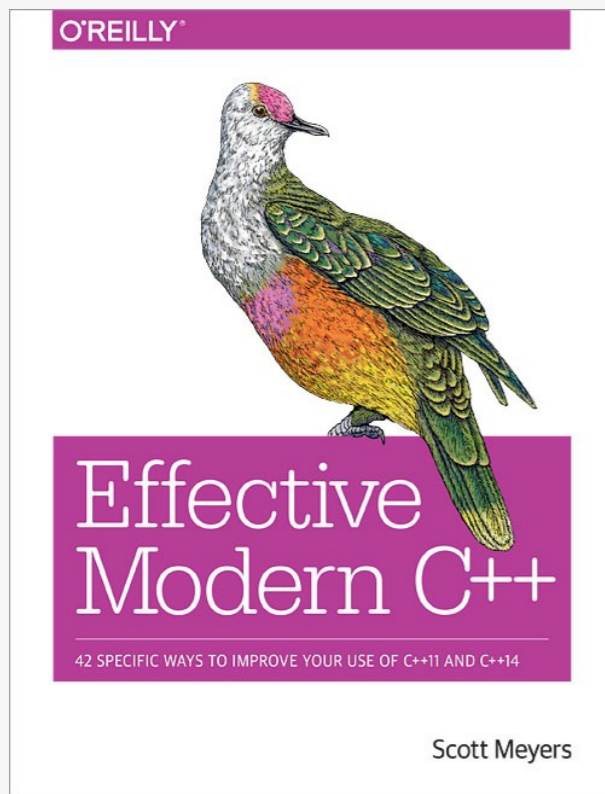
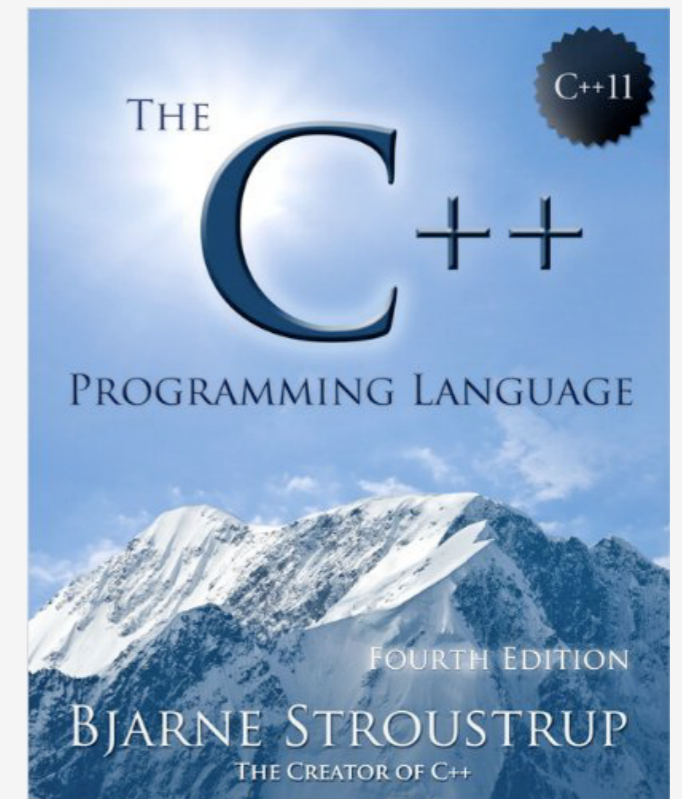
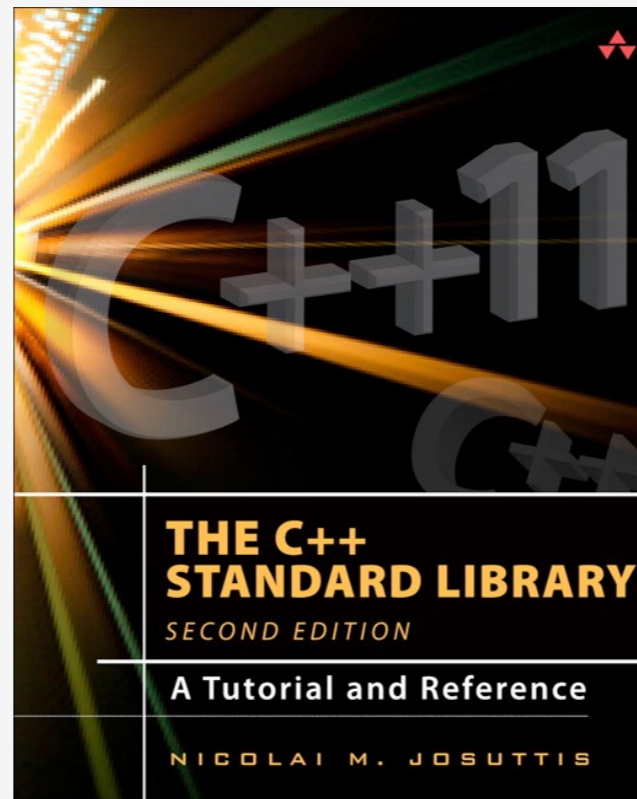
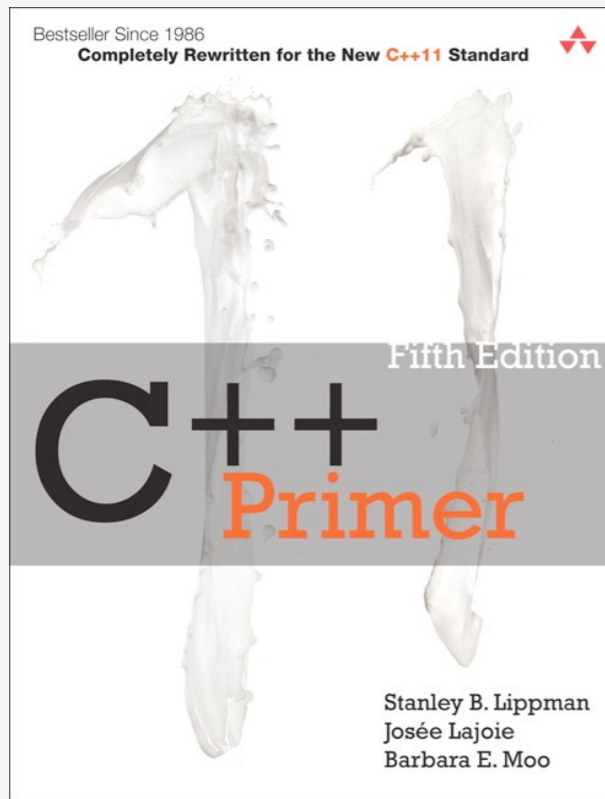
3. Object Oriented C++ 1

4. Generic C++ and the STL


5. Object Oriented C++ 2

```
int main(int argc, char *argv[])  
{  
    int result = calculate_foo();  
    return 0;  
}
```

```
int main(int argc, char *argv[])  
{  
    FooCalculator foo;  
    int result = foo.calculate();  
    return 0;  
}
```




C++ Bookshelf



Get Started! Tour Core Guidelines Super-FAQ Standardization About

News, Status & Discussion about Standard C++


Follow All Posts 

The home of Standard C++ on the web — news, status and discussion about the C++ standard on all compilers and platforms.

FEATURES



Programmers Q&A



Turn ON syntax

Top
In: Introduction
In: Philosophy

C++ Core Guidelines

August 3, 2020

Editors:

- [Bjarne Stroustrup](#)
- [Herb Sutter](#)

Fork me on GitHub



cppreference.com Create account

Page Discussion View View source History

 **CppCon 2020**
It's the annual, week-long gathering for the entire C++ community. [Register now!](#)

C++ reference
C++98, C++03, C++11, C++14, C++17, C++20

[Compiler support \(11, 14, 17, 20\)](#) [Concepts library \(C++20\)](#) [Iterators library](#)

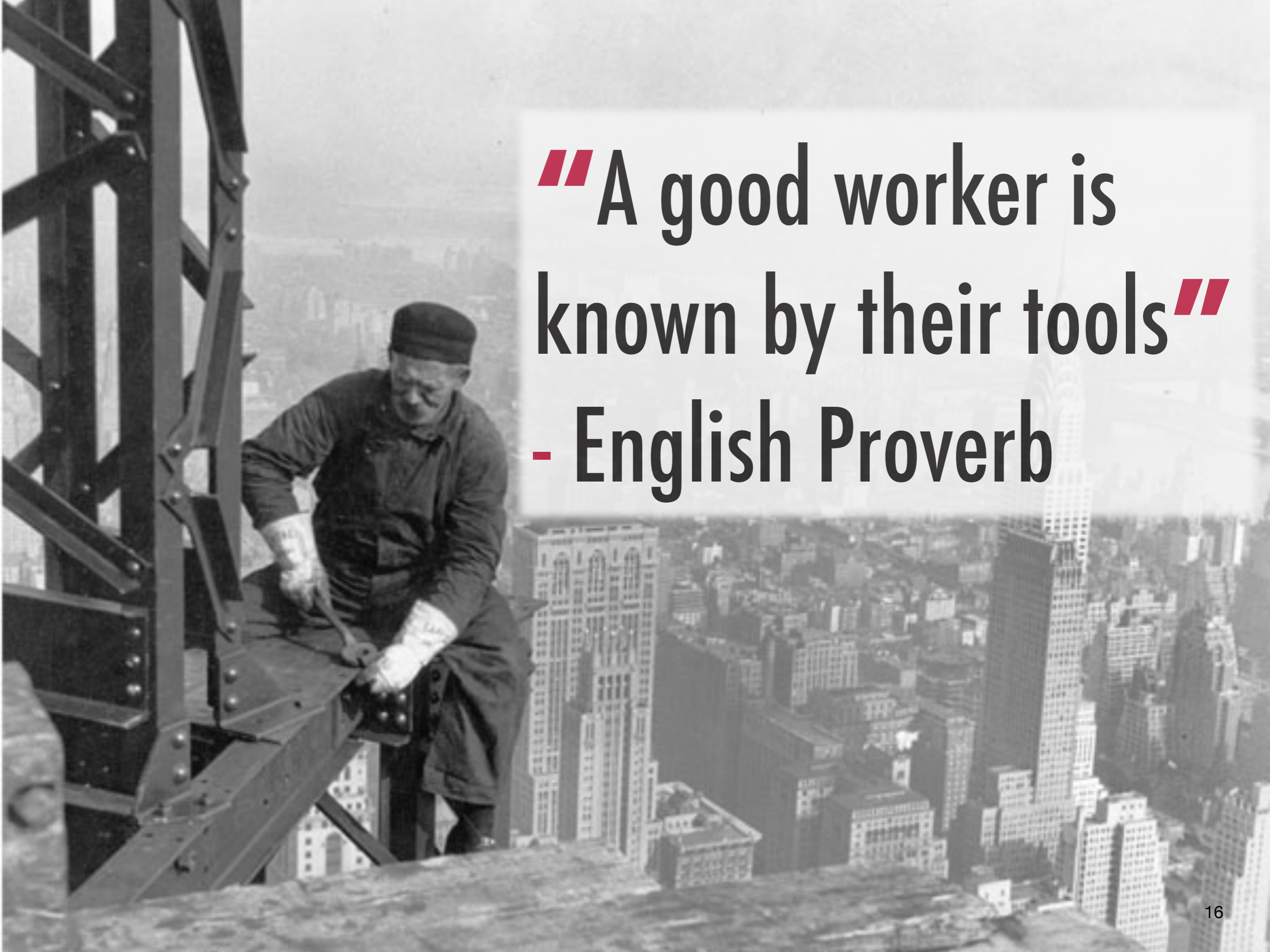


C++ Online Resources



“If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization.”

- Gerald Weinberg



“A good worker is known by their tools”
- English Proverb

1. Storing code and changes
(**Git**)



2. Automating builds
(**Compiler/CMake**)



3. Testing (**CMake/Catch**)

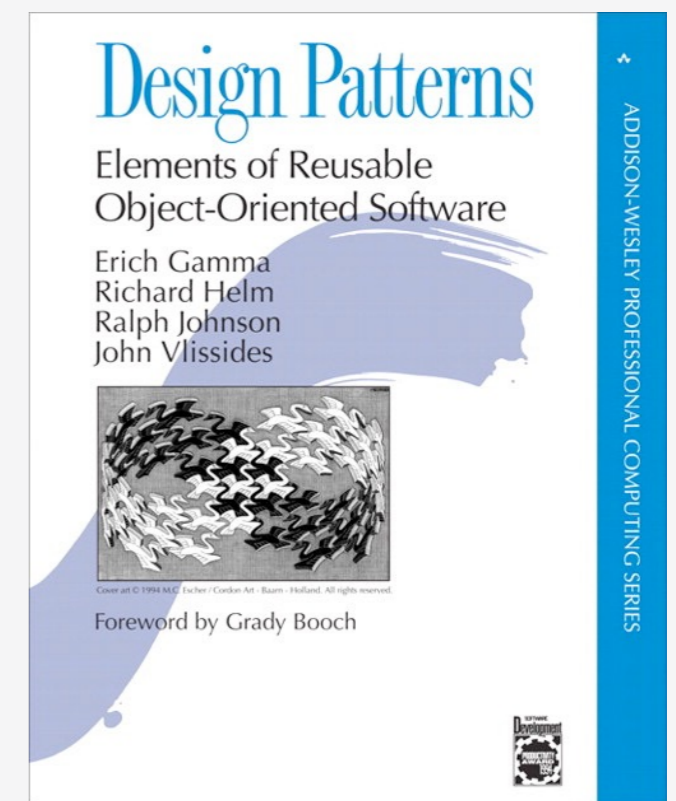
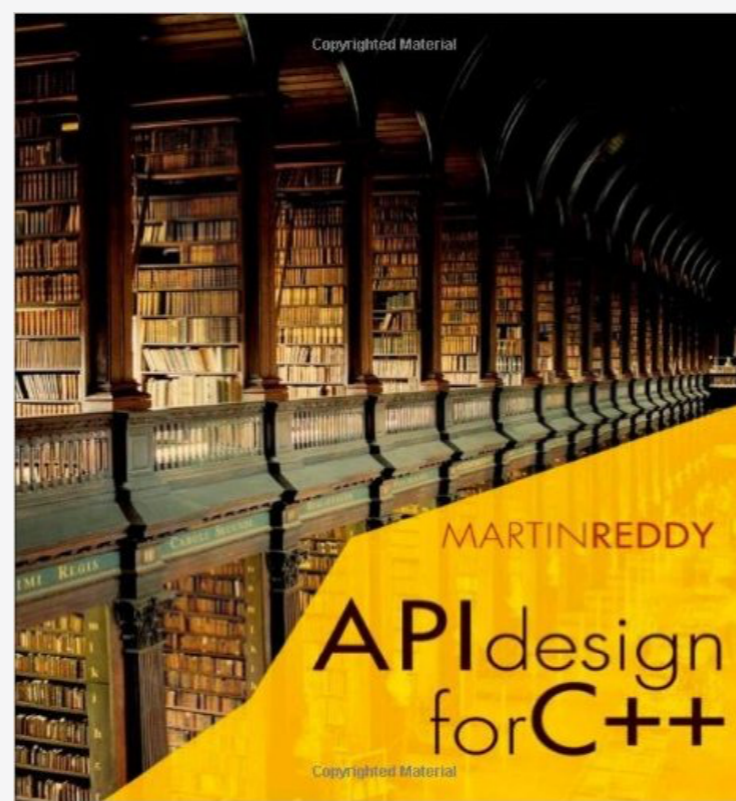
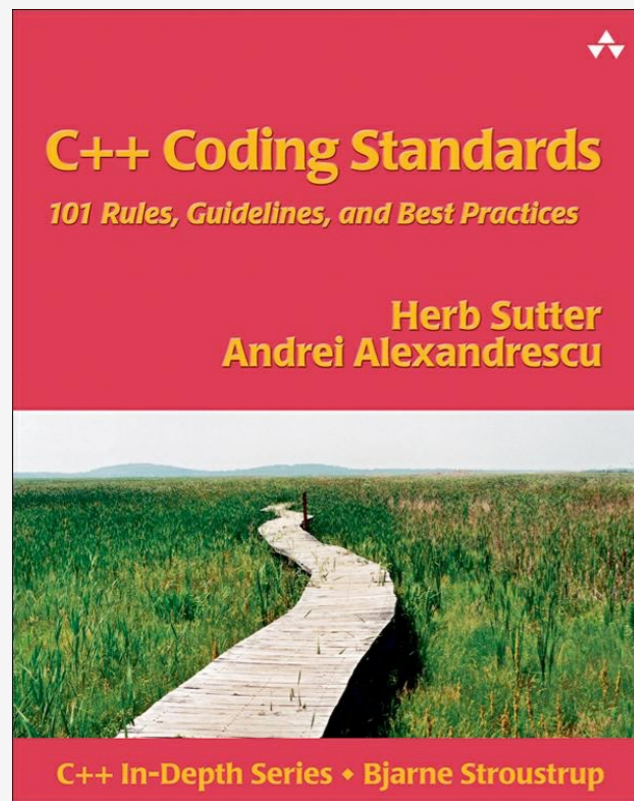
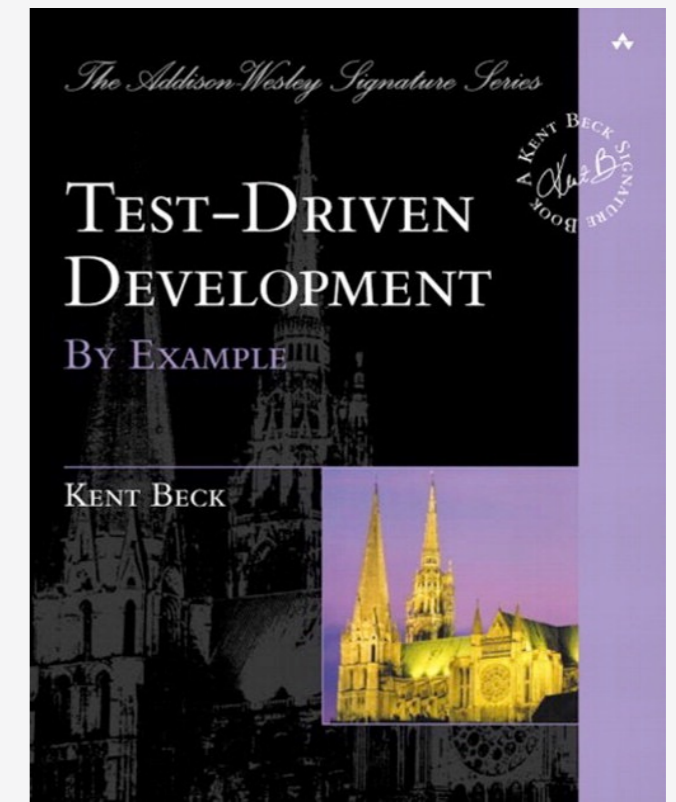
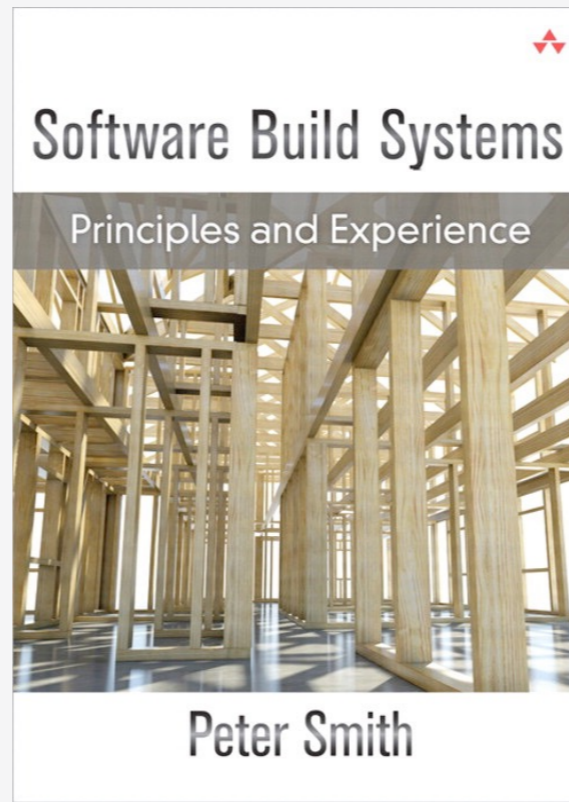
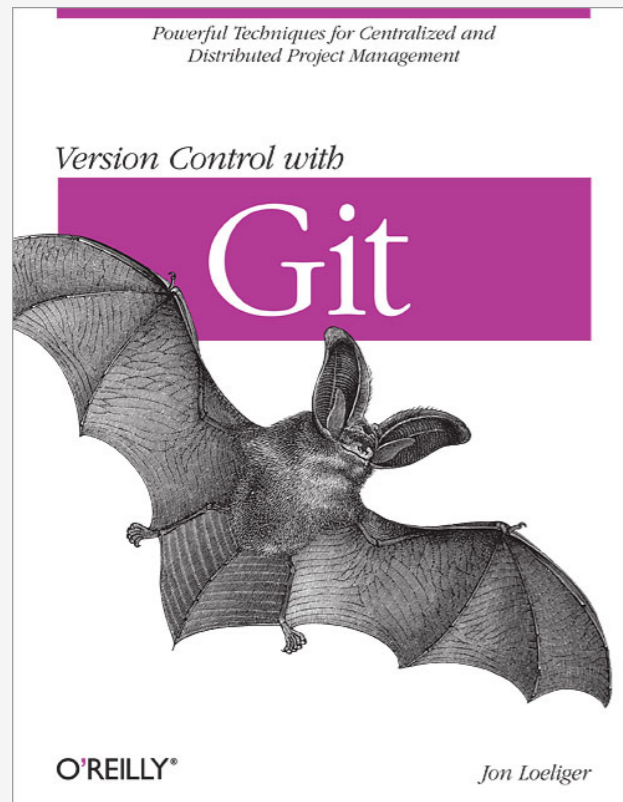


4. Documenting code and programs (**Doxygen**)



5. *Software Design*

Software Development Stream



Software Development Bookshelf

git --distributed-is-the-new-centralized

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



github

SOCIAL CODING

CMake » latest release (3.18.3) Documentation » next | index

Table of Contents

- Introduction
- Command-Line Tools
- Interactive Dialogs
- Reference Manuals
- Guides
- Release Notes
- Index and Search

Next topic

[cmake\(1\)](#)

This Page

[Show Source](#)

Quick search

Introduction

CMake is a tool to manage building of source code. Originally, CMake was designed as a generator for various dialects of `Makefile`, today CMake generates modern buildsystems such as `Ninja` as well as project files for IDEs such as Visual Studio and Xcode.

CMake is widely used for the C and C++ languages, but it may be used to build source code of other languages too.

People encountering CMake for the first time may have different initial goals. To learn how to build a source code package downloaded from the internet, start with the [User Interaction Guide](#). This will detail the steps needed to run the `cmake(1)` or `cmake-gui(1)` executable and how to choose a generator, and how to complete the build.

The [Using Dependencies Guide](#) is aimed at developers wishing to get started using a third-party library.

stackoverflow

Developer Q&A

Apple Developer News Discover Design Develop Distribute Support Account



msdn

Home Library Learn Downloads Support Comm

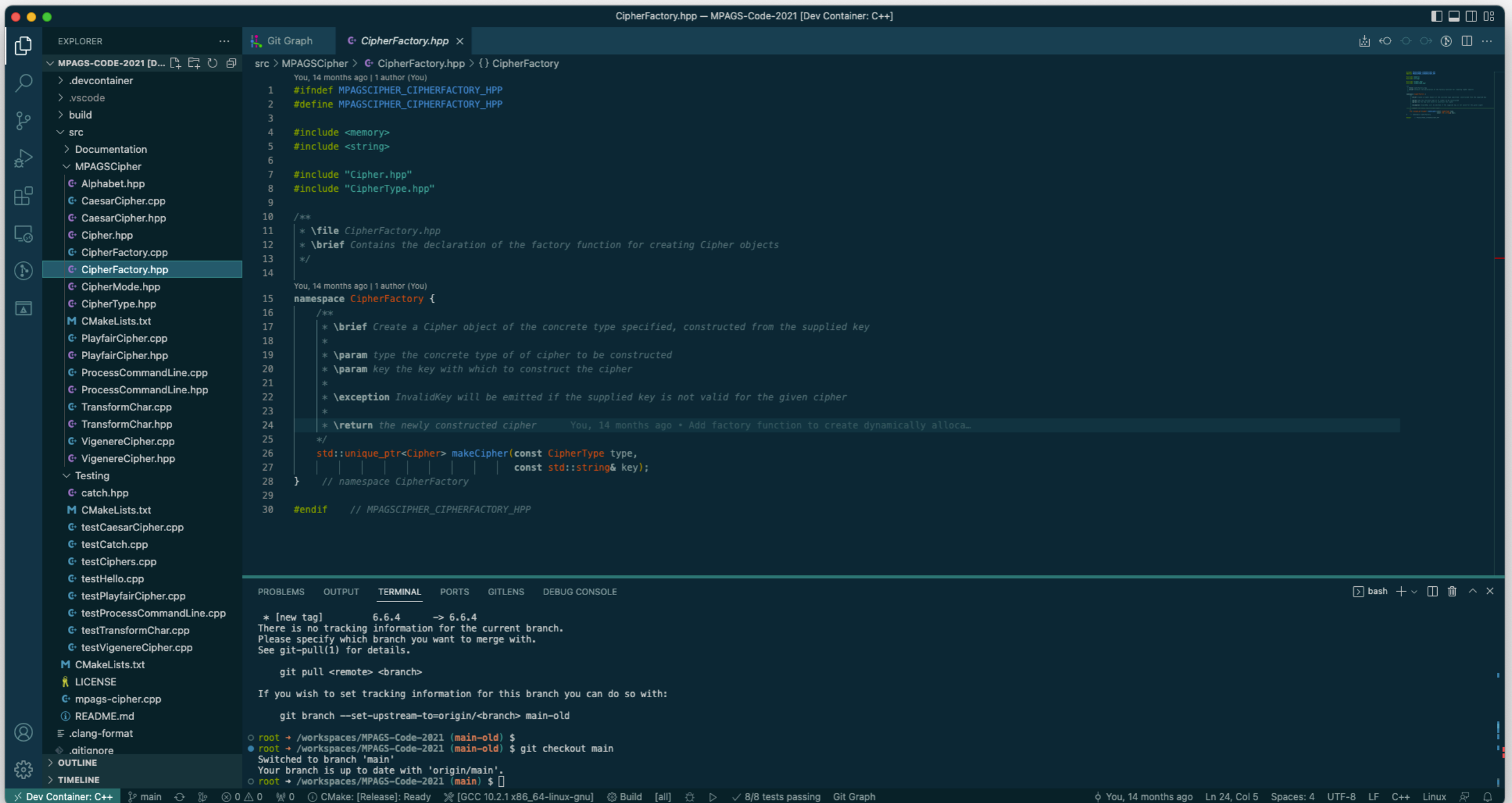
PORTALS:

Platforms

Tasks

explore platfo

Software Development Online Resources



Getting Started

```
$ mpags-cipher --help
```

```
> questions?
```

```
> comments?
```