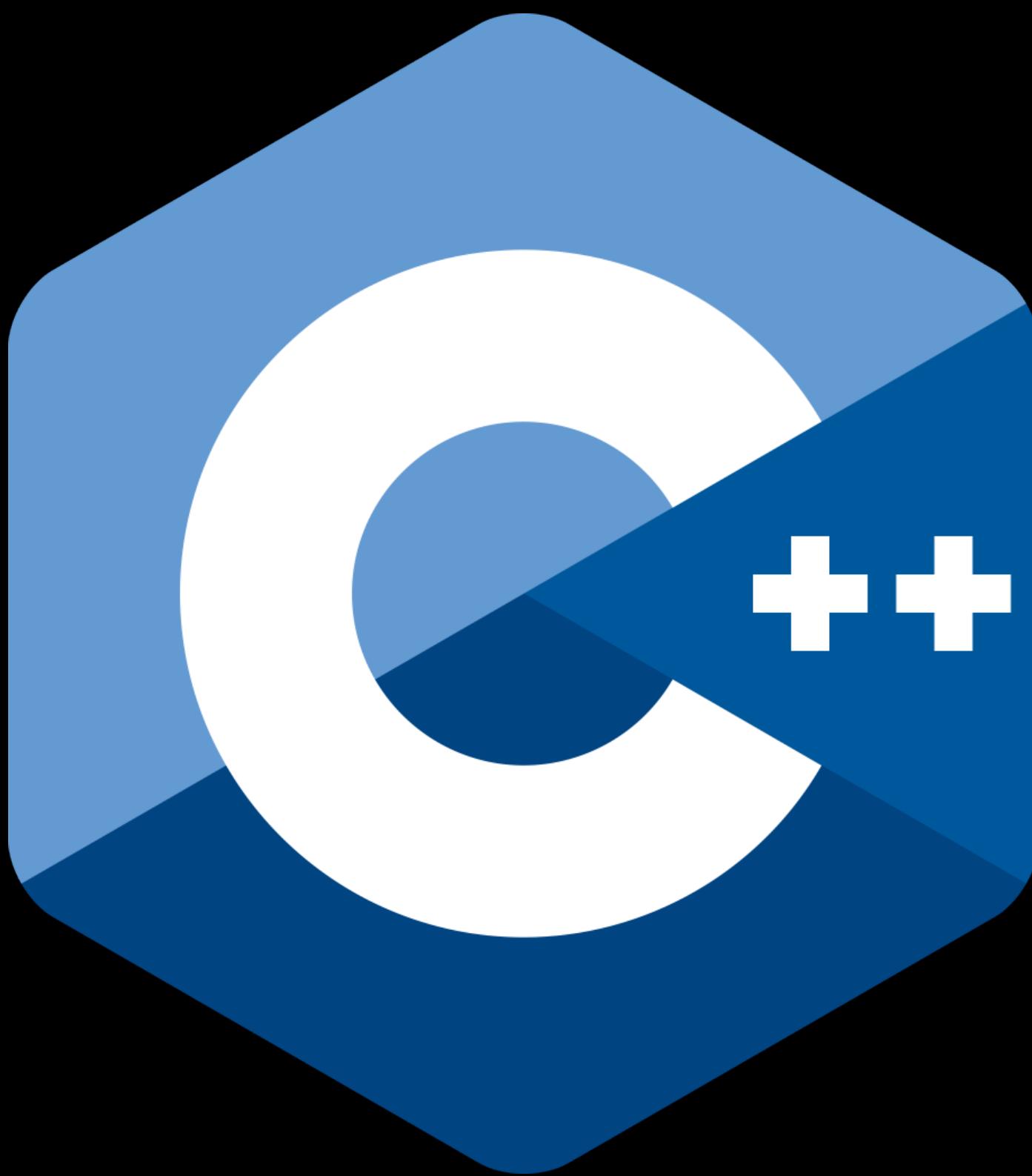


# Modern C++ Course



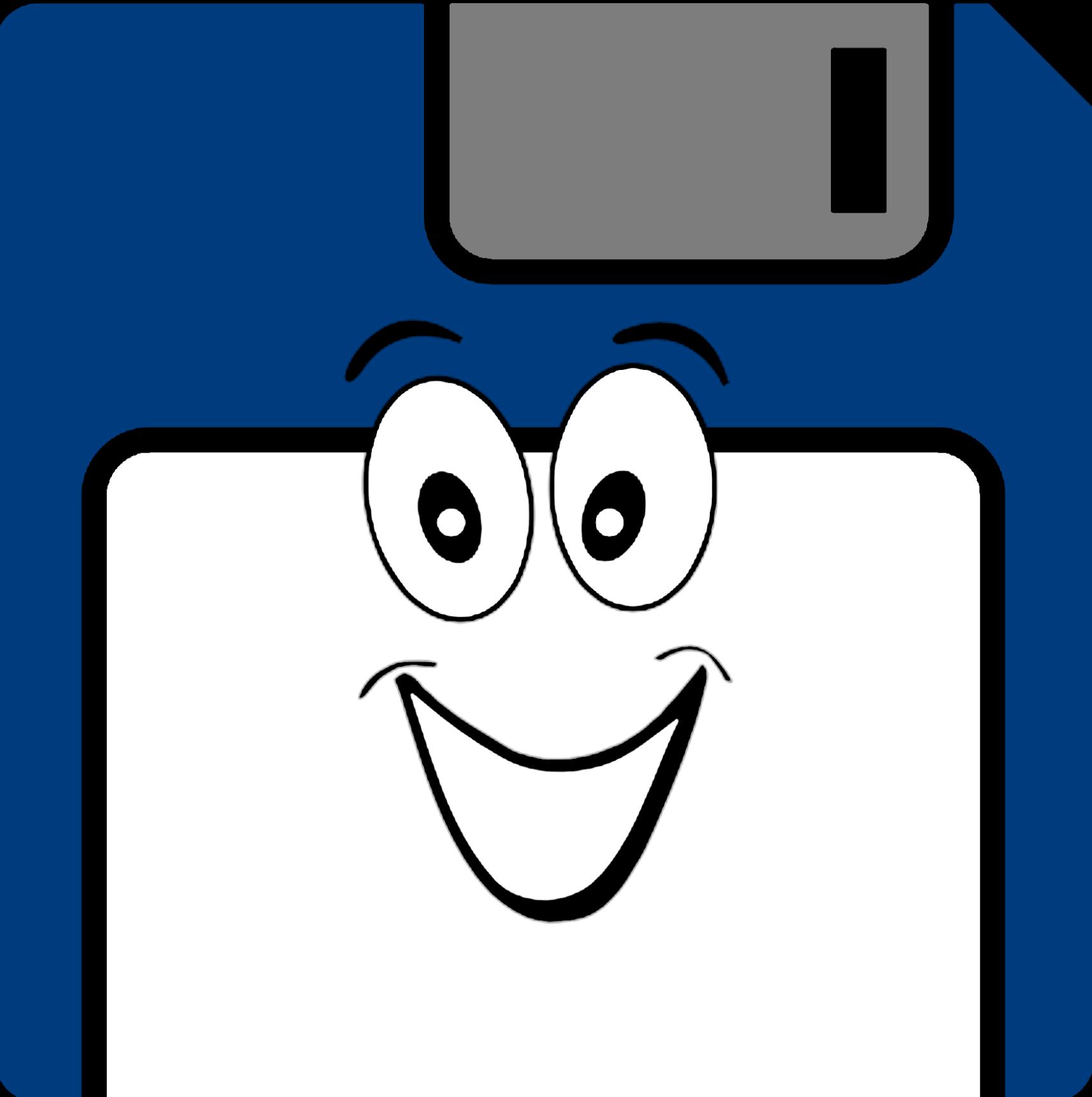
# Who am I ?

## Gammasoft

Gammasoft aims to make c++ fun again.

## About

- Gammasoft is the nickname of Yves Fiumefreddo.
- More than thirty years of passion for high technology especially in development (c++, c#, objective-c, ...).
- Object-oriented programming is more than a mindset.
- more info see my GitHub : <https://github.com/gammasoft71>



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# What this course is?



# What this course is?

- This course aimed at inexperienced developers as well as experienced developers with no knowledge of Modern C++.



# What this course is?

- This course aimed at inexperienced developers as well as experienced developers with no knowledge of Modern C++.
- This course covers concepts from C++98 to C++26, while keeping modern C++ in mind.

# What this course is?

- This course aimed at inexperienced developers as well as experienced developers with no knowledge of Modern C++.
- This course covers concepts from C++98 to C++26, while keeping modern C++ in mind.
- The corresponding C++ version is indicated for each concept and feature.

# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



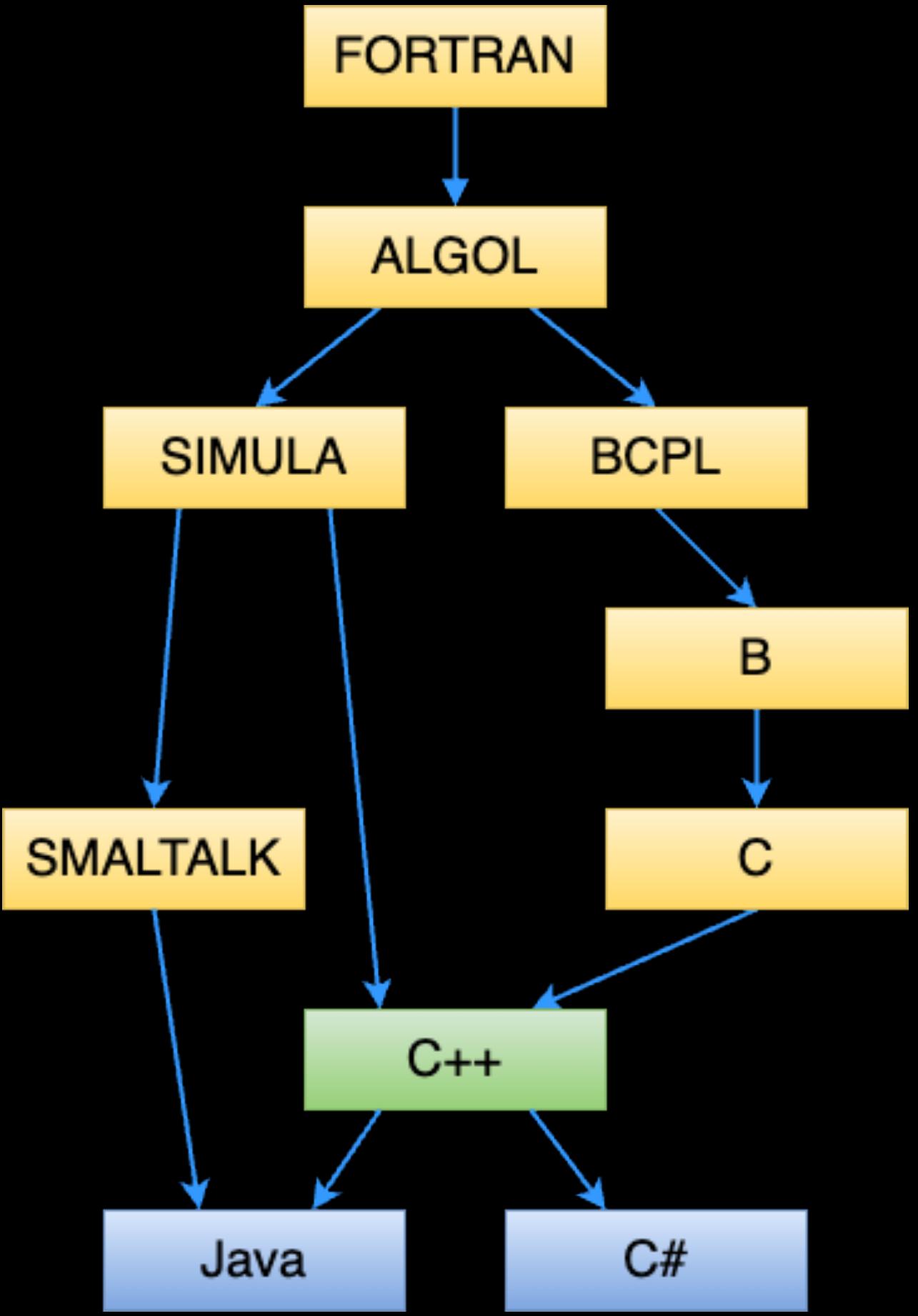
# Bjarne Stroustrup

Bjarne Stroustrup, a Danish computer scientist, is the creator of C++.



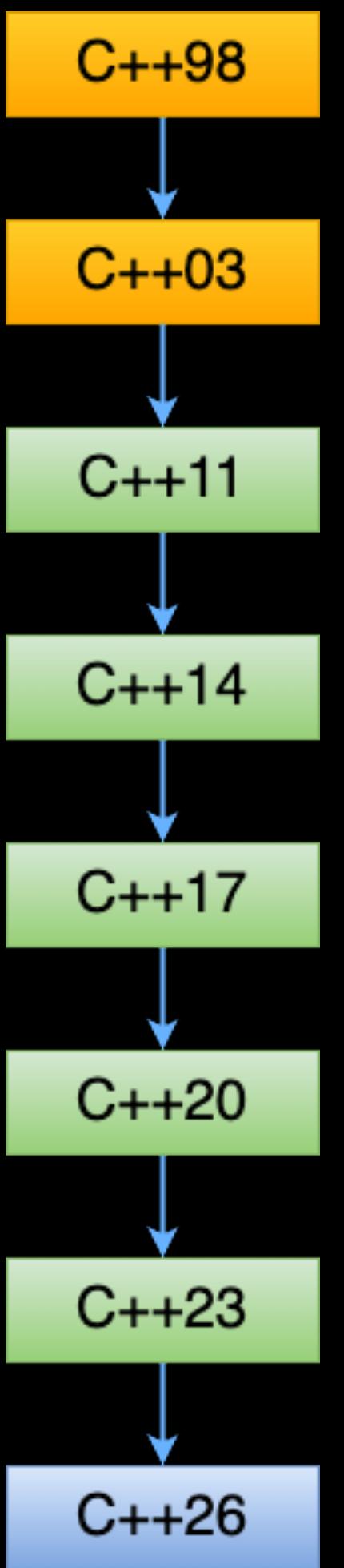
# C++ language

- In 1979, Bjarne Stroustrup began work on "**C with Classes**", the predecessor to "C++".
- In 1982, Stroustrup started to develop a successor to "C with Classes", which he named "C++".
- In 1985, the first edition of The C++ Programming Language was released
- In 1998, C++98 was released, standardizing the language



# C++ Standards

- 1998 | 14882:1998 | C++98
- 2003 | 14882:2003 | C++03
- 2011 | 14882:2011 | C++11
- 2014 | 14882:2014 | C++14
- 2017 | 14882:2017 | C++17
- 2020 | 14882:2020 | C++20
- 2024 | 14882:2024 | C++23
- TBA | | C++26



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Adapted to large projects



# Adapted to large projects

- Statically and strongly typed

# Adapted to large projects

- Statically and strongly typed
- Object oriented



# Adapted to large projects

- Statically and strongly typed
- Object oriented
- Widely used (and taught)



# Adapted to large projects

- Statically and strongly typed
- Object oriented
- Widely used (and taught)
- Many available libraries



# Adapted to large projects

- Statically and strongly typed
- Object oriented
- Widely used (and taught)
- Many available libraries
- Many available documentations and examples on the net



# Fast



# Fast

- unlike some other languages (Java, C#, Python, ...) C++ is compiled to native machine code



# Fast

- unlike some other languages (Java, C#, Python, ...) C++ is compiled to native machine code
- close to hardware when needed



# What we get?



# What we get?

- The most powerful language



# What we get?

- The most powerful language
- The most complicated one



# What we get?

- The most powerful language
- The most complicated one
- The most error prone?



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Introduction

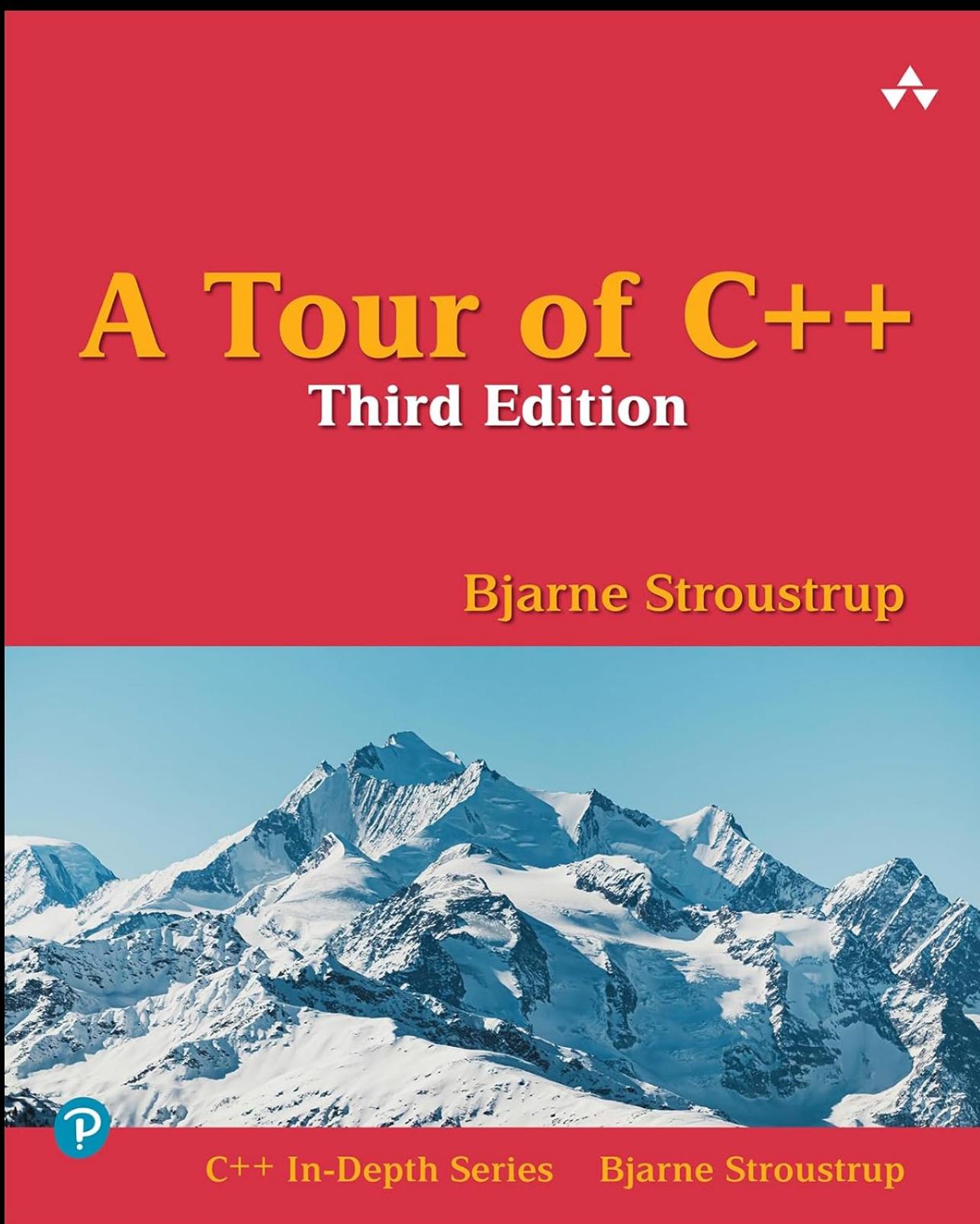
- What this course is?
- History
- Why using C++?
- References
- C++ development tools



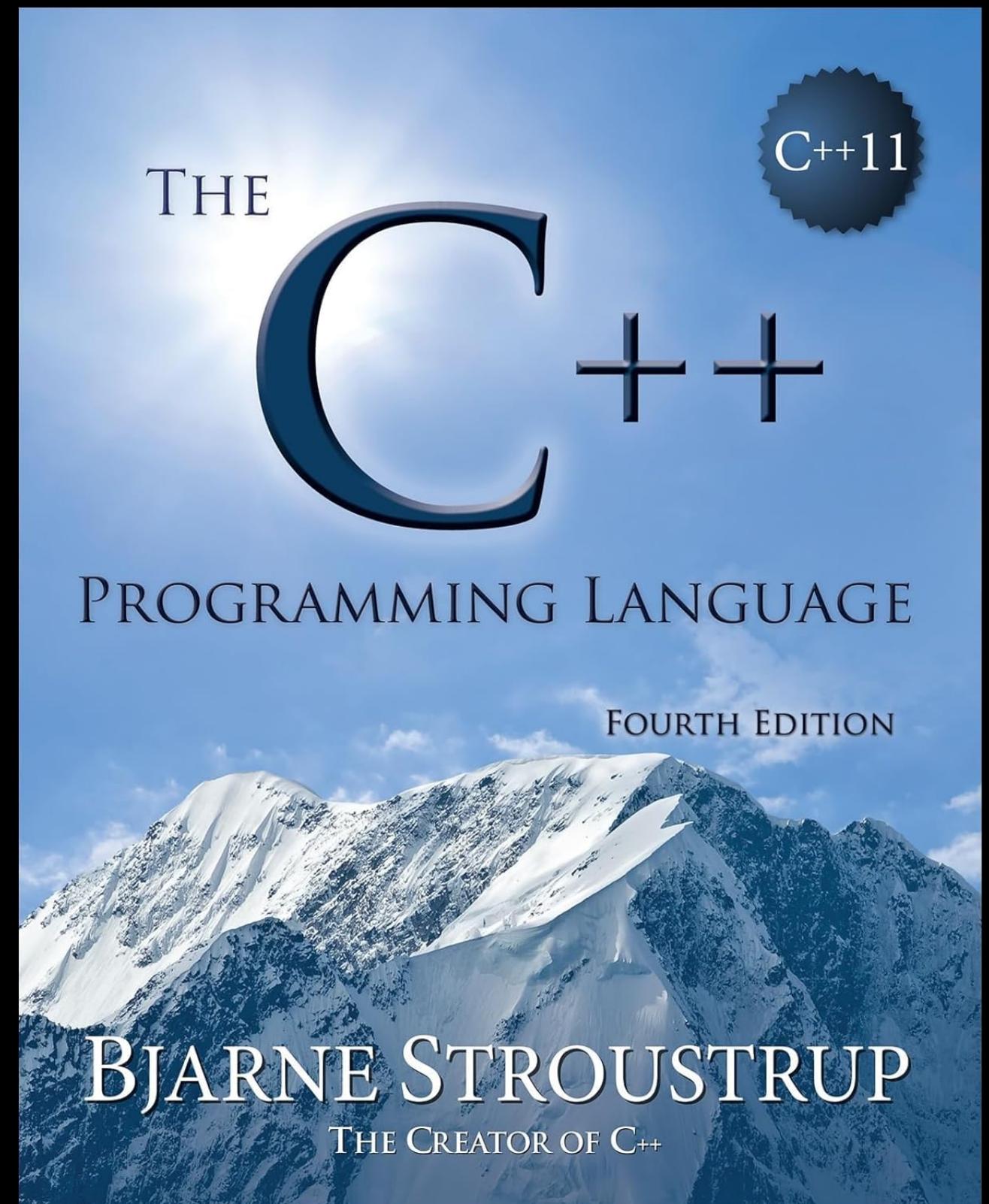
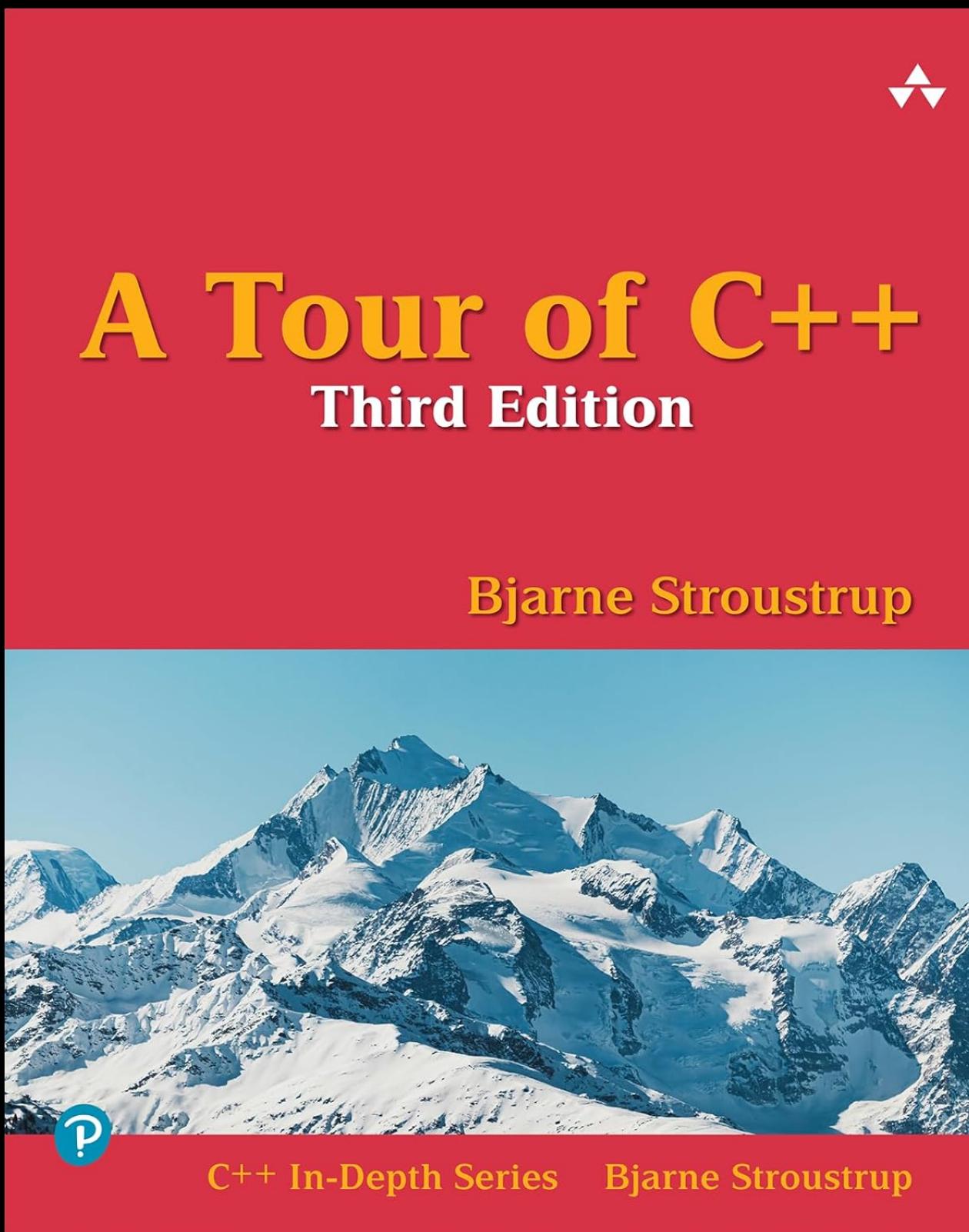
# Books



# Books



# Books



# Useful links



# Useful links

- <https://isocpp.org>



# Useful links

- <https://isocpp.org>
- <https://cppreference.com>

# Useful links

- <https://isocpp.org>
- <https://cppreference.com>
- <https://github.com>



# Useful links

- <https://isocpp.org>
- <https://cppreference.com>
- <https://github.com>
- <https://stackoverflow.com>



# Useful links

- <https://isocpp.org>
- <https://cppreference.com>
- <https://github.com>
- <https://stackoverflow.com>
- <https://www.programmingalgorithms.com>



# Useful links

- <https://isocpp.org>
- <https://cppreference.com>
- <https://github.com>
- <https://stackoverflow.com>
- <https://www.programmingalgorithms.com>
- And more...

# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Compilers

C++	GCC	Clang	MSVC	Apple Clang
11	$\geq 4.8$	$\geq 3.3$	$\geq 19$	$\geq 3.3$
14	$\geq 4.9$	$\geq 3.4$	$\geq 19$	$\geq 3.3$
17	$\geq 7.3$	$\geq 5$	$\geq 19.01$	$\geq 10$
20	$\geq 11$	$\geq 19$	$\geq 19.29$	$\geq 15$
23	$\geq 15$	$\geq 19$	$\geq 19.40$	$\geq 15$



# Integrated Development Environment (IDE)



# Integrated Development Environment (IDE)

-  Microsoft Visual Studio (Windows)

# Integrated Development Environment (IDE)

-  Microsoft Visual Studio (Windows)
-  Xcode (macOS)

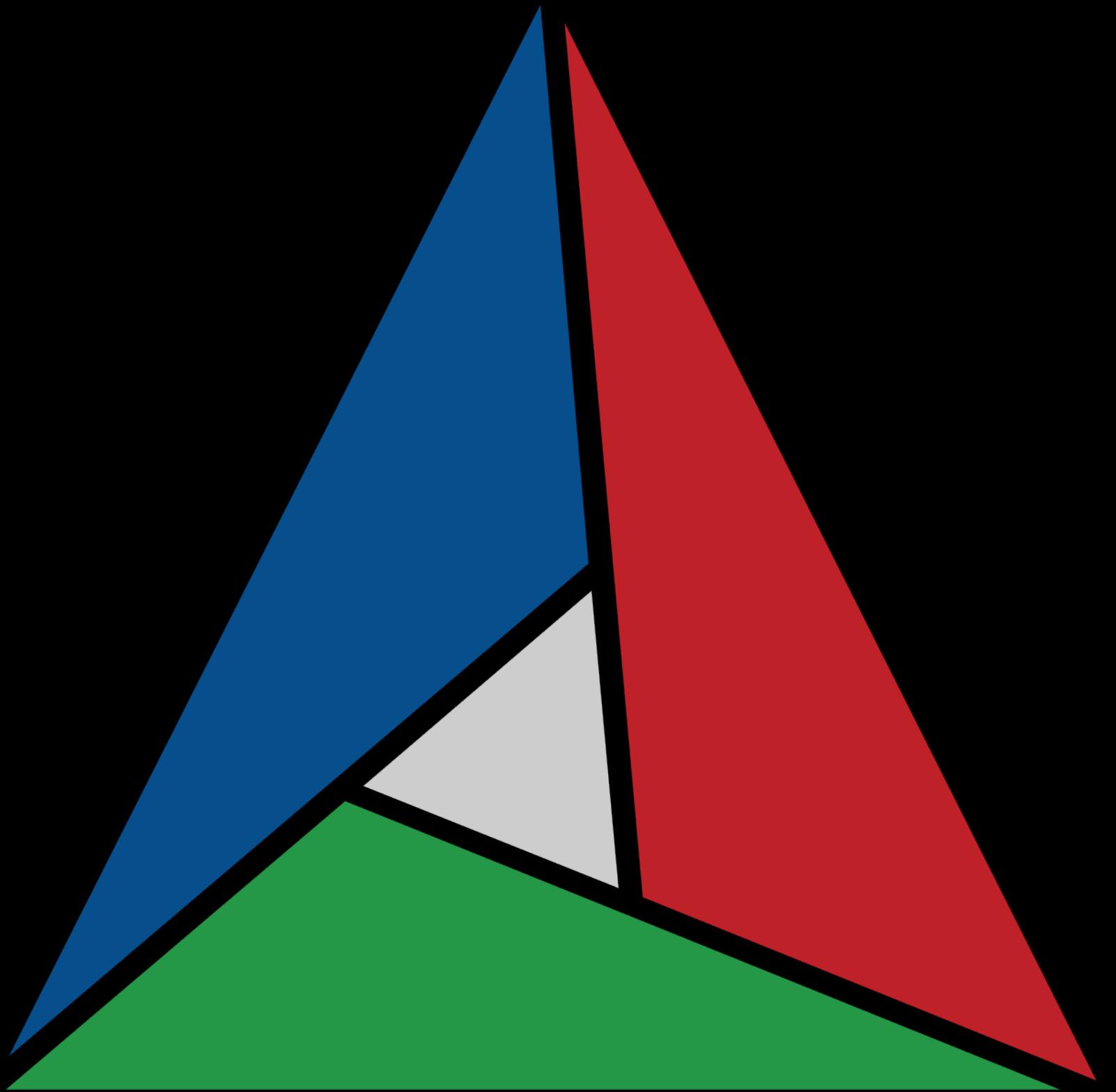
# Integrated Development Environment (IDE)

-  Microsoft Visual Studio (Windows)
-  Xcode (macOS)
-  Microsoft Visual Studio Code (Windows, macOS and Linux)

# Integrated Development Environment (IDE)

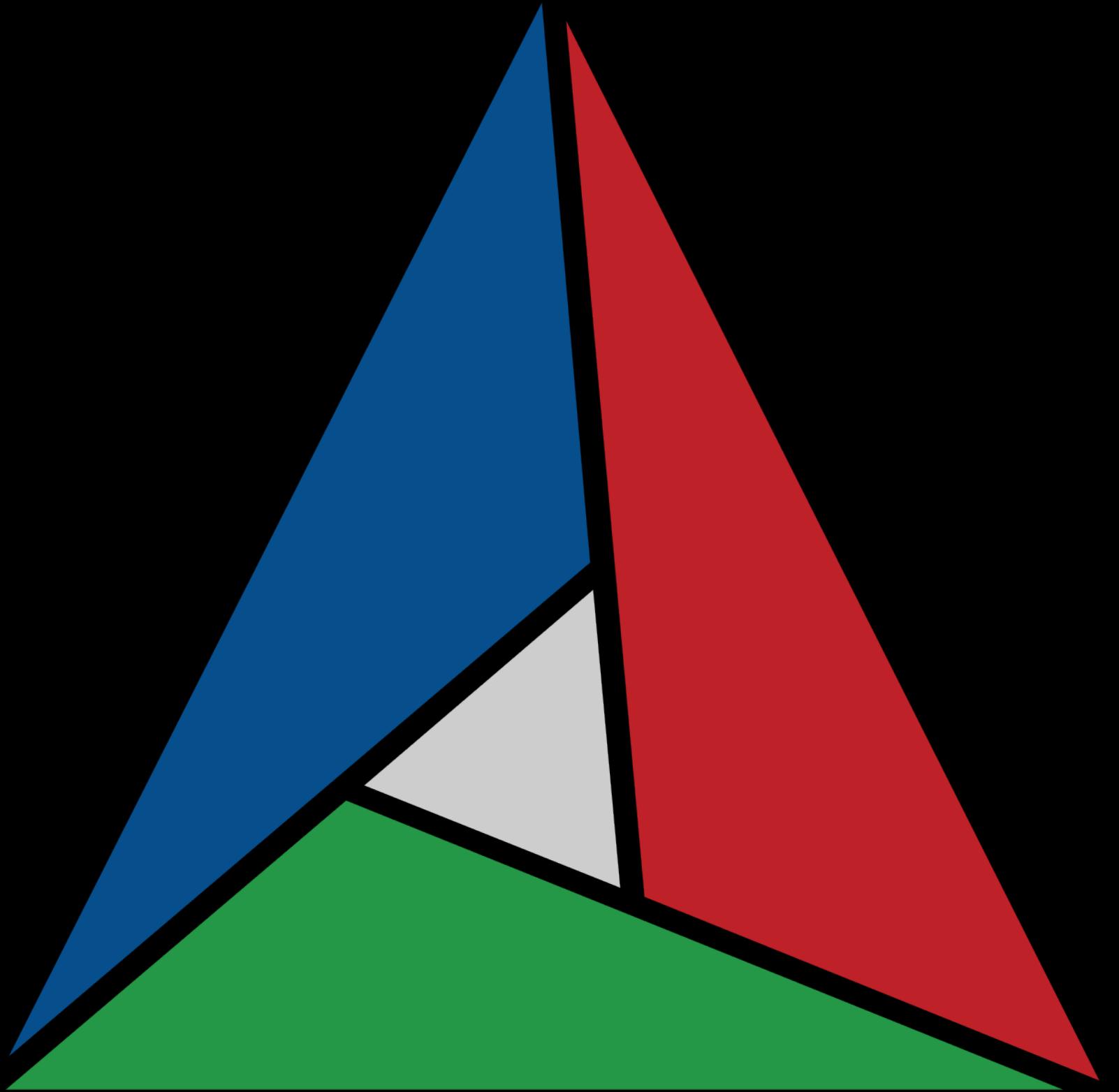
-  Microsoft Visual Studio (Windows)
-  Xcode (macOS)
-  Microsoft Visual Studio Code (Windows, macOS and Linux)
-        and others

# CMake



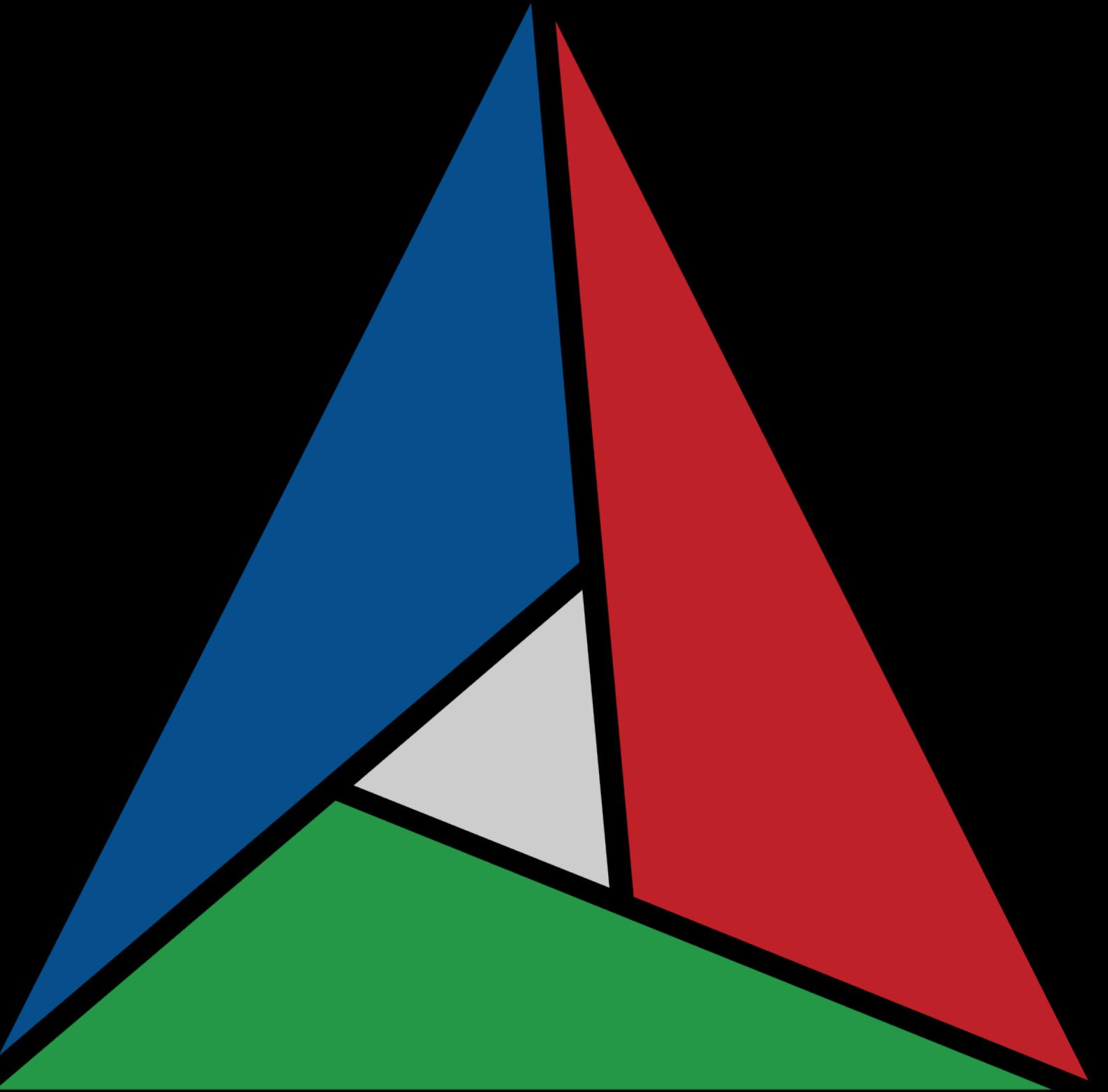
# CMake

- Single source builds on multiple platforms



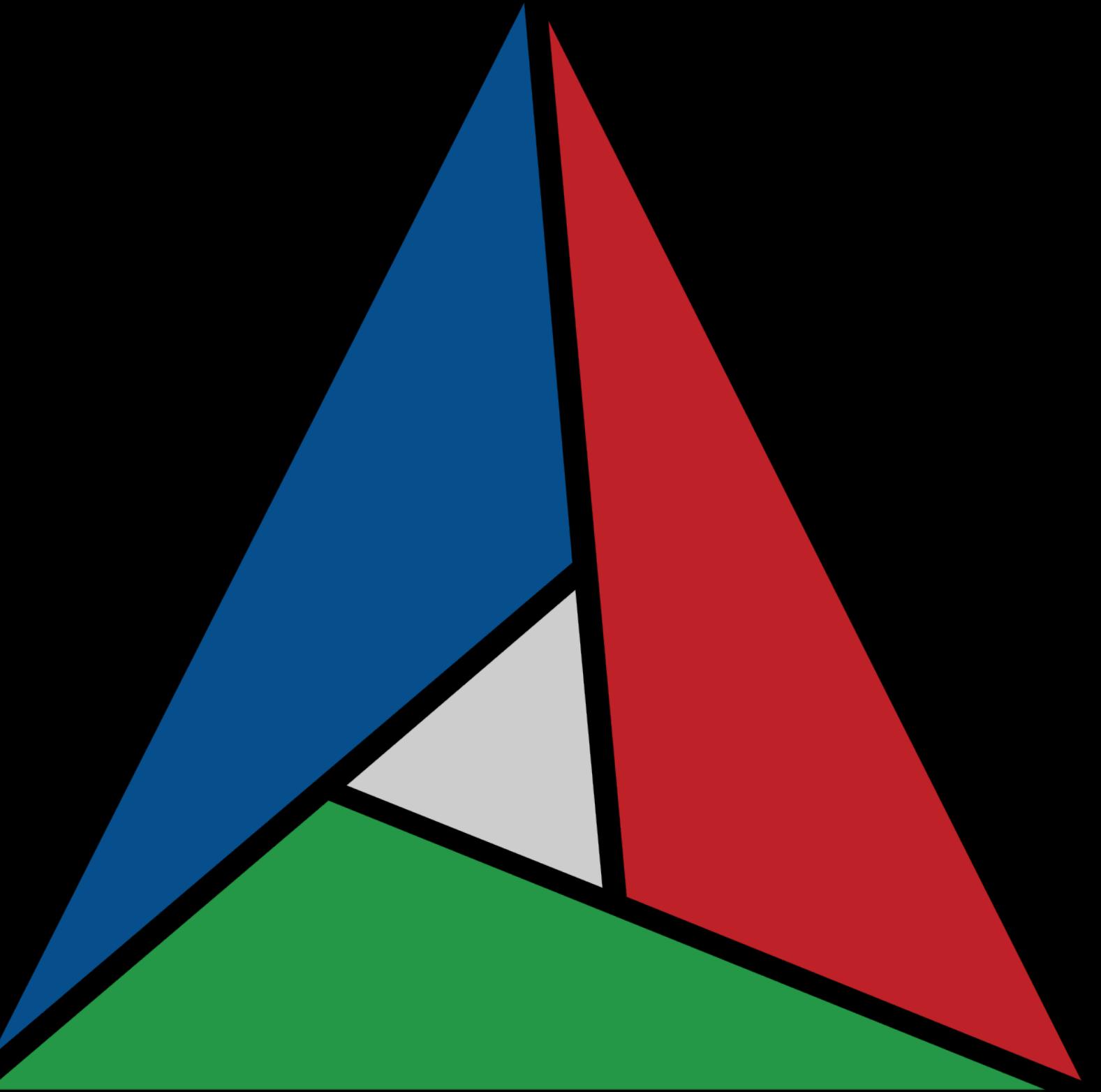
# CMake

- Single source builds on multiple platforms
- Accurate dependencies and minimal rebuilds



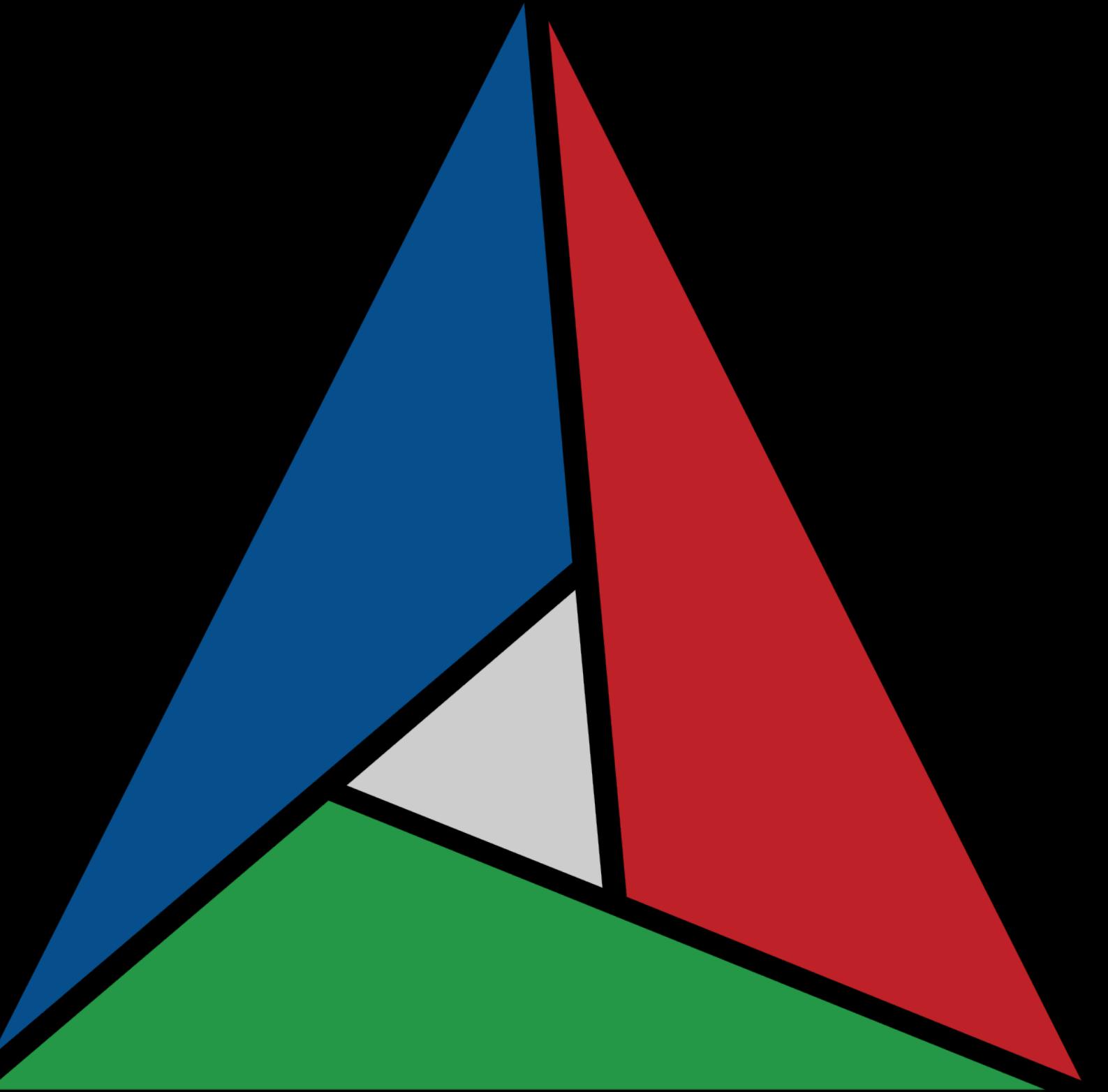
# CMake

- Single source builds on multiple platforms
- Accurate dependencies and minimal rebuilds
- Out-of-source builds



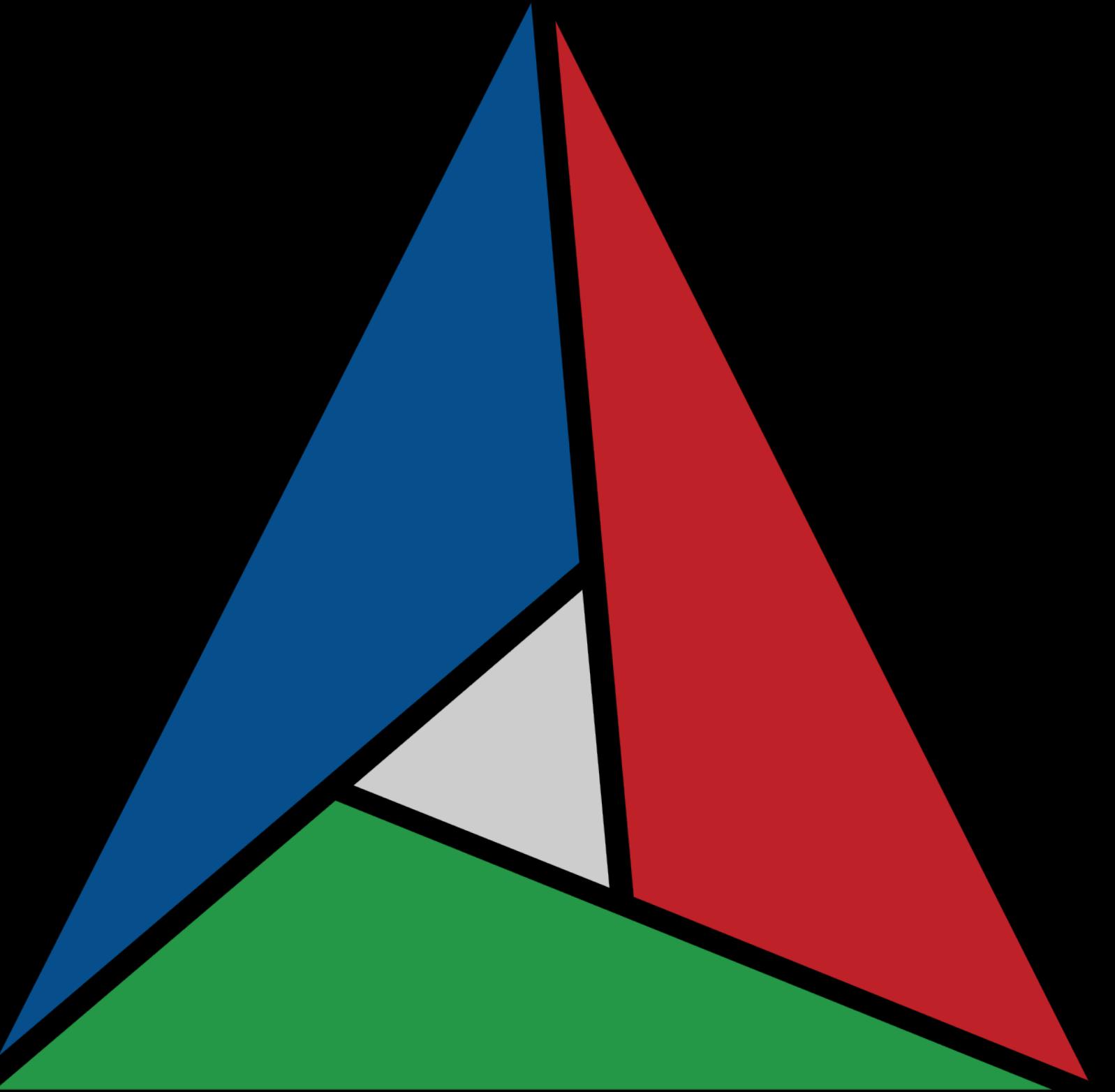
# CMake

- Single source builds on multiple platforms
- Accurate dependencies and minimal rebuilds
- Out-of-source builds
- Cross-platform packaging system



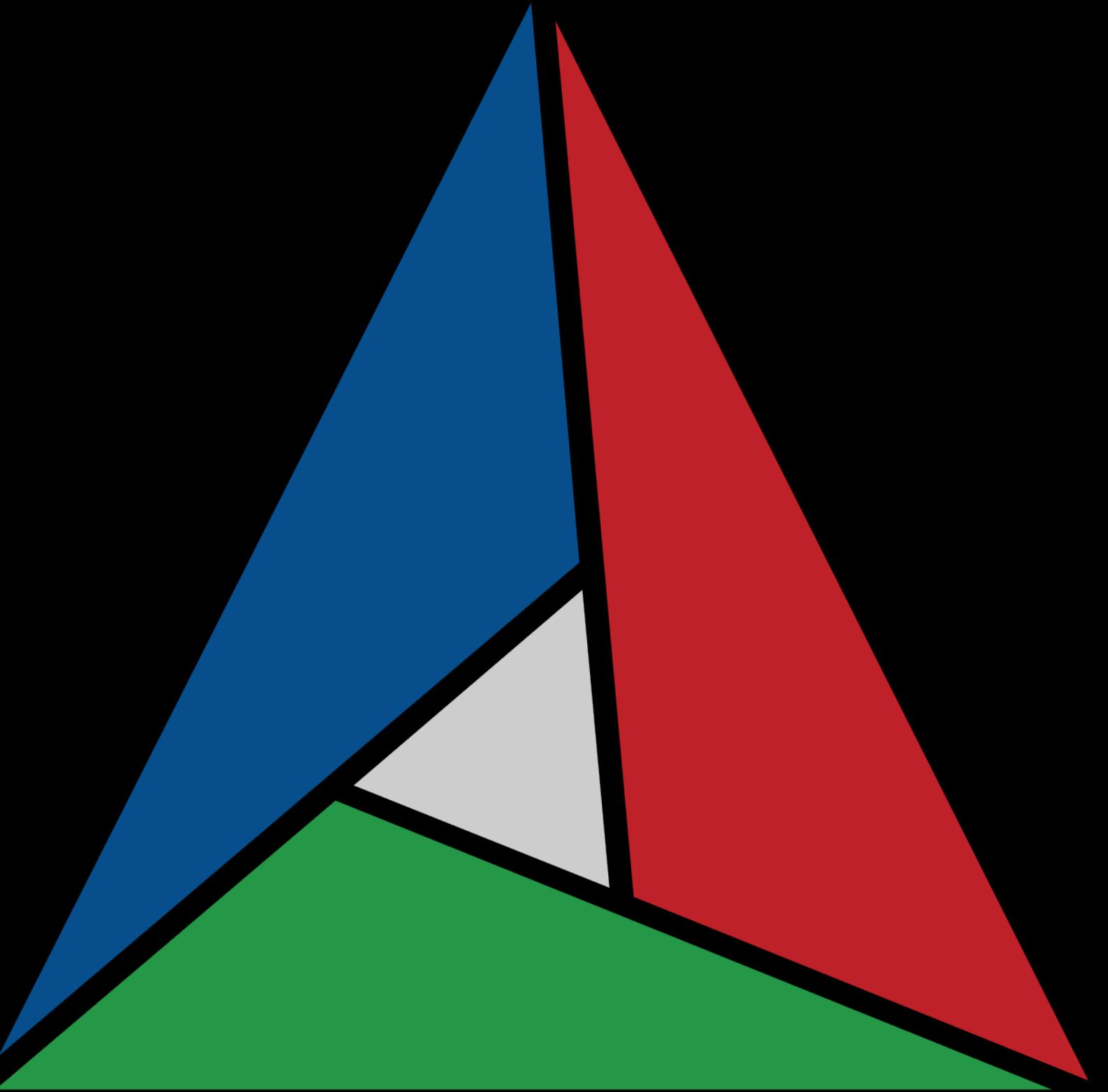
# CMake

- Single source builds on multiple platforms
- Accurate dependencies and minimal rebuilds
- Out-of-source builds
- Cross-platform packaging system
- Cross-platform testing system

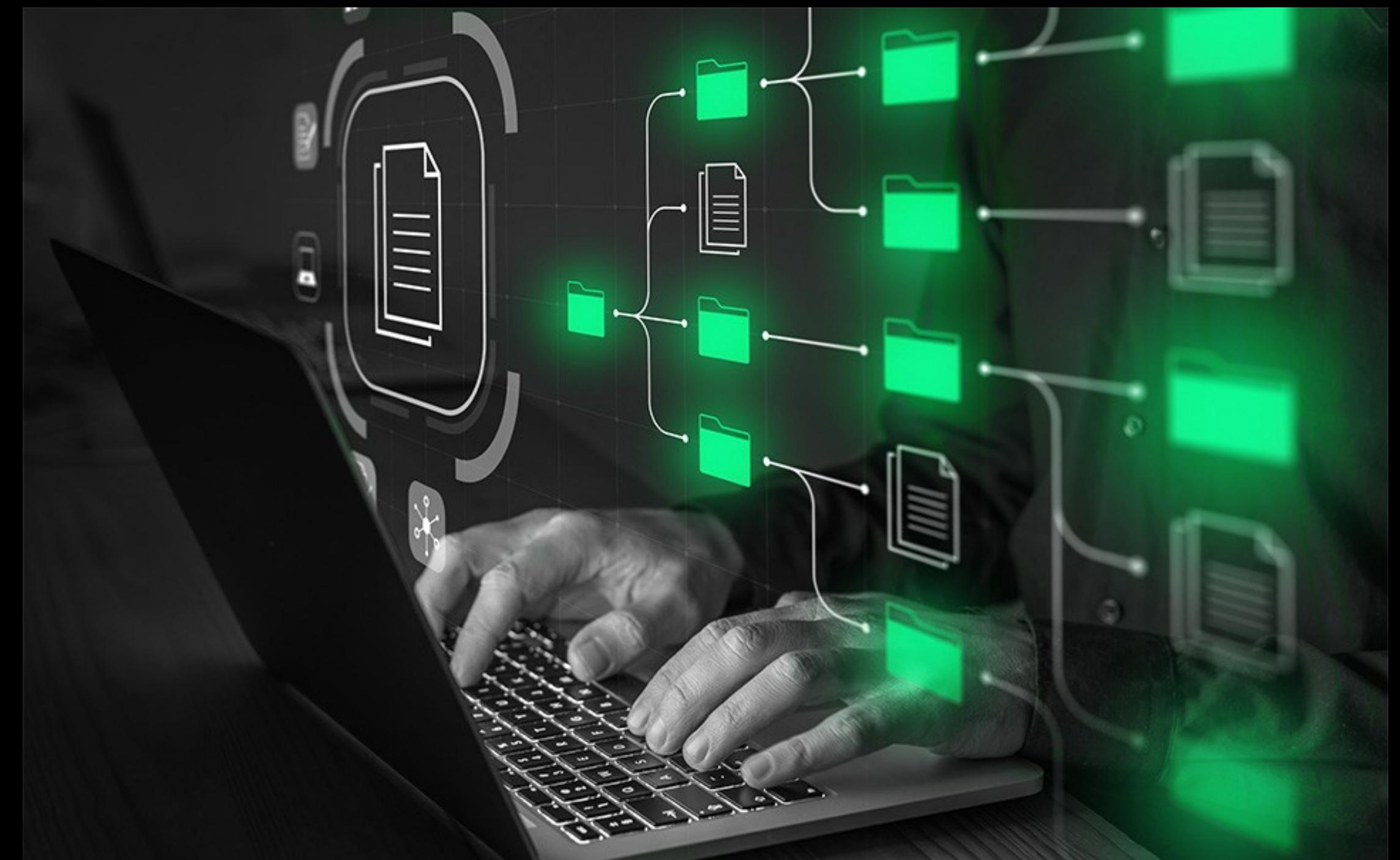


# CMake

- Single source builds on multiple platforms
- Accurate dependencies and minimal rebuilds
- Out-of-source builds
- Cross-platform packaging system
- Cross-platform testing system
- And more...

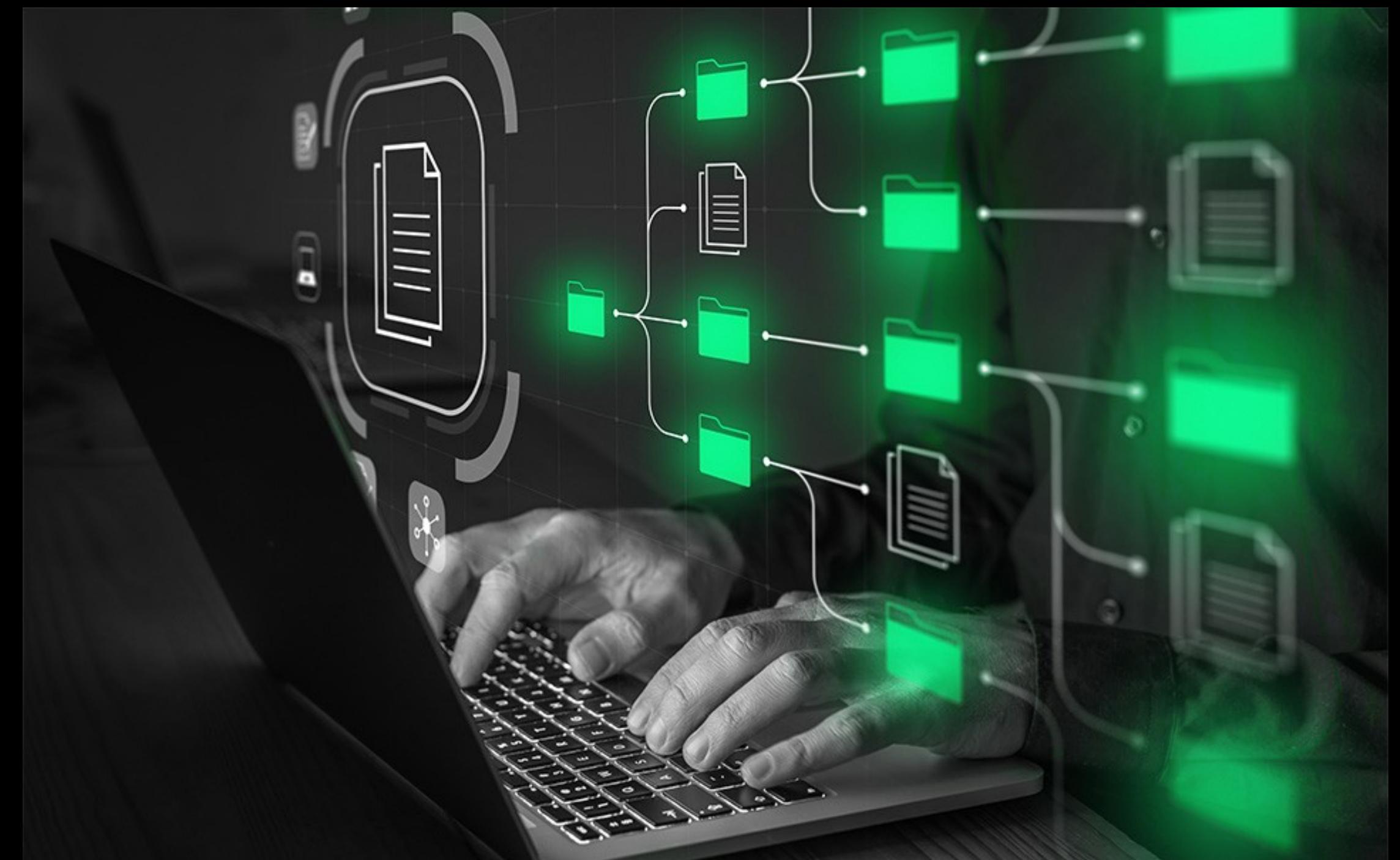


# Configure your development environment



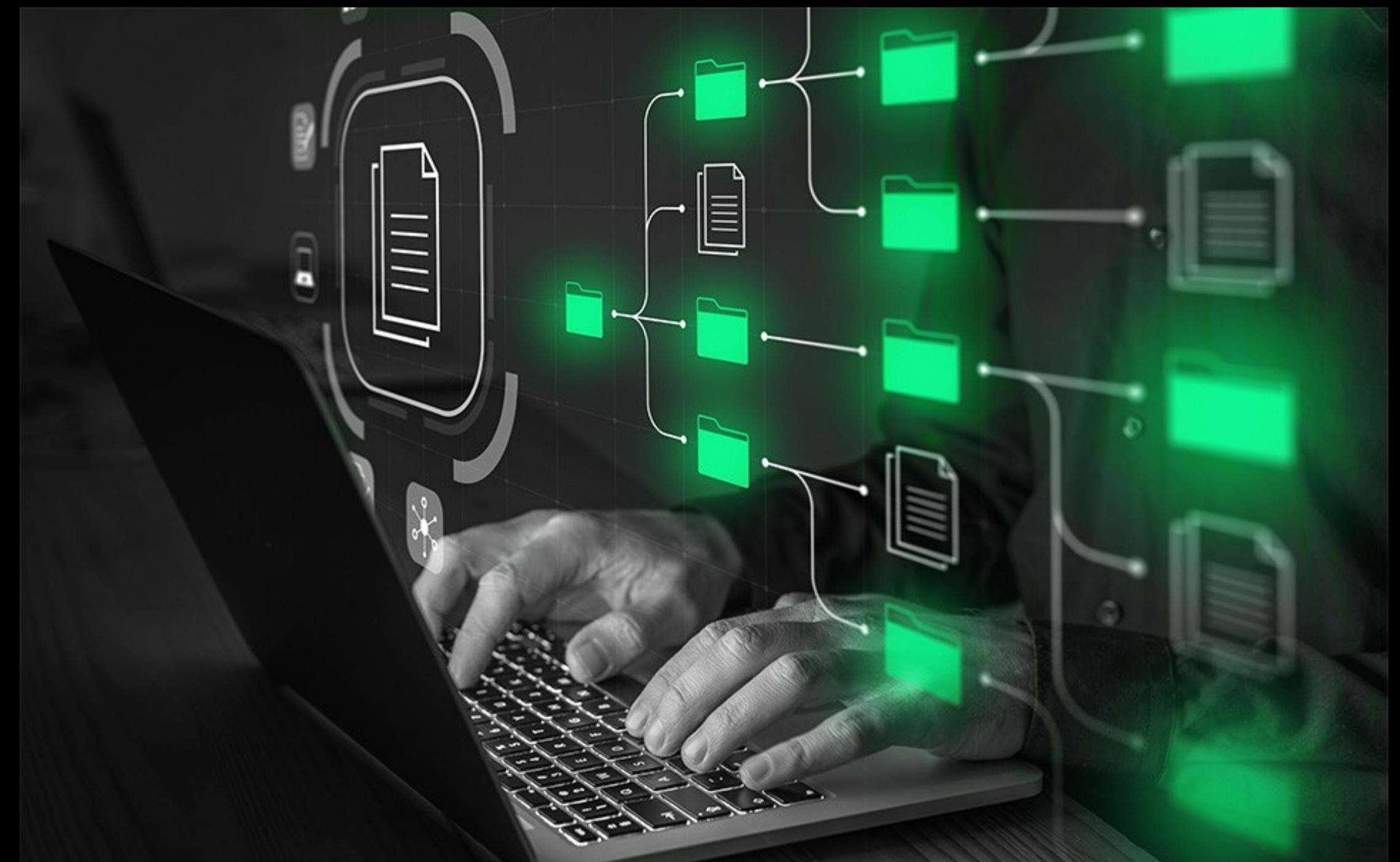
# Configure your development environment

- Install compiler



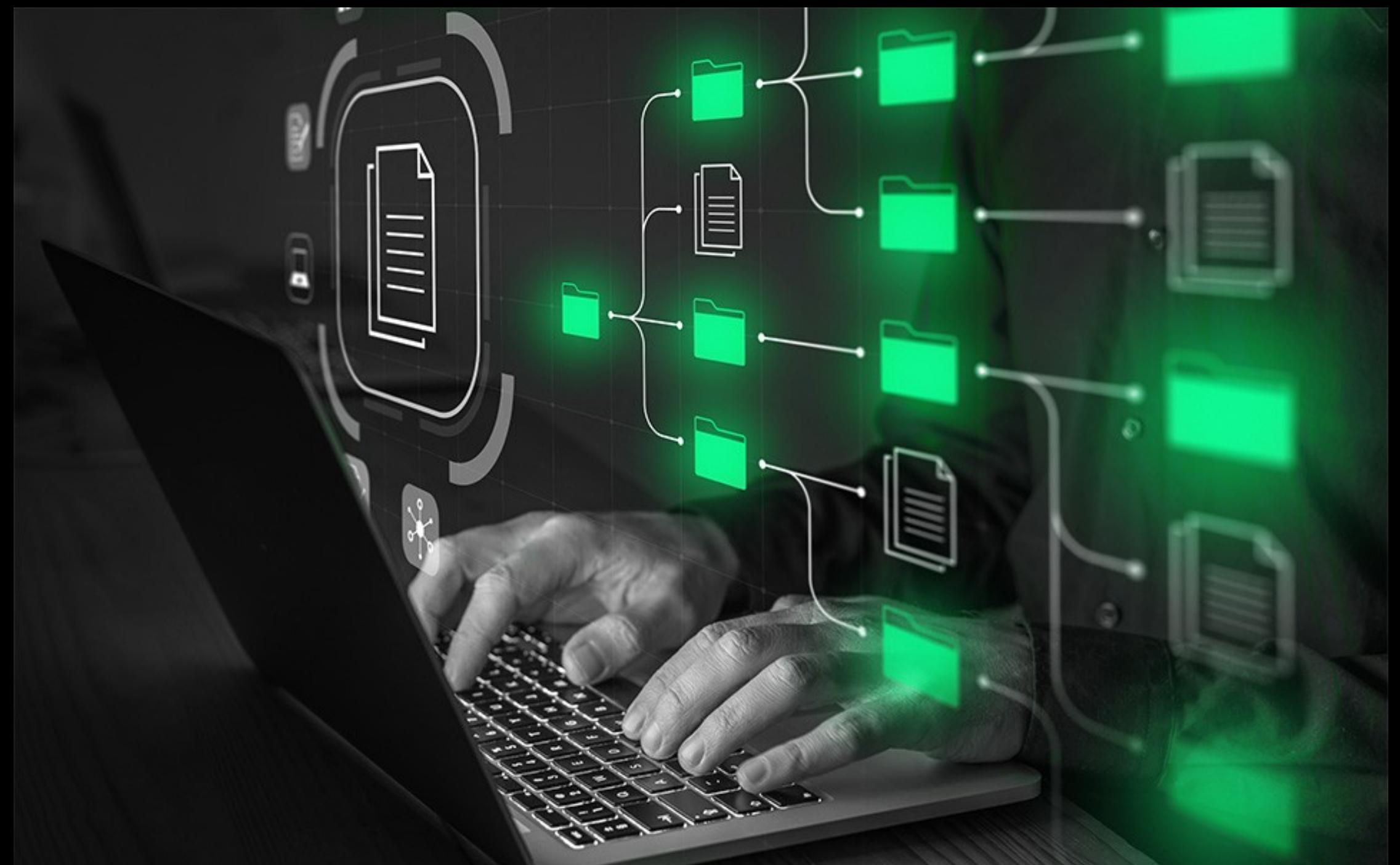
# Configure your development environment

- Install compiler
- Install IDE (according to OS)



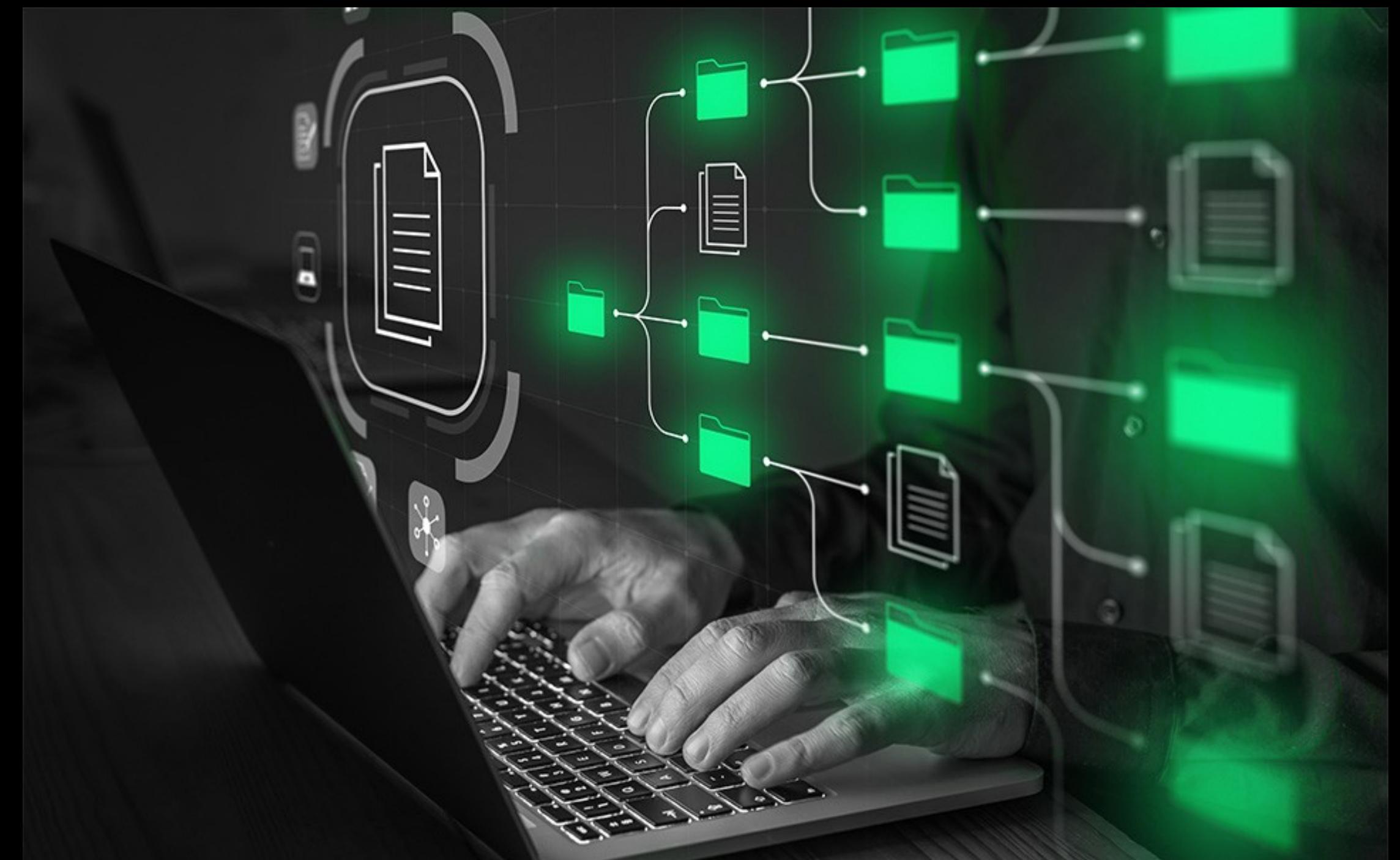
# Configure your development environment

- Install compiler
- Install IDE (according to OS)
- Install CMake



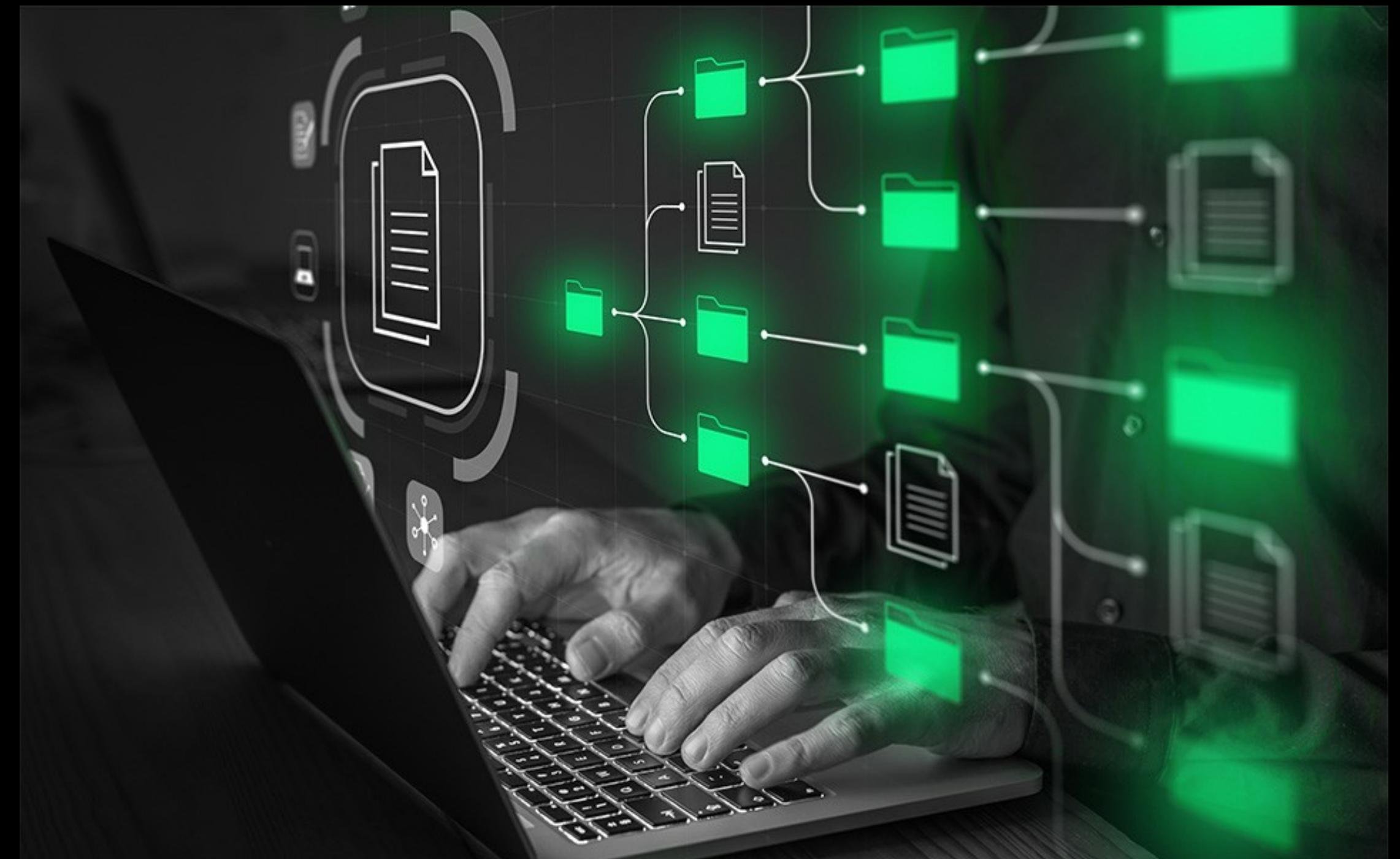
# Configure your development environment

- Install compiler
- Install IDE (according to OS)
- Install CMake
- Install SCM like git (not mandatory)



# Configure your development environment

- Install compiler
- Install IDE (according to OS)
- Install CMake
- Install SCM like git (not mandatory)
- Subscribe to GitHub (not mandatory)



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Introduction

- What this course is?
- History
- Why using C++?
- References
- C++ development tools



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# Outline

1. Introduction
2. Language Basics
3. Object Oriented Programming (OOP)
4. Core Modern C++
5. Modern C++ Expert
6. Advanced Programming



# End

