High Level Programming

Introduction to C++

Stefano Quer and Alessandro Savino
Dipartimento di Automatica e Informatica
Politecnico di Torino

C++

- C++ is a high-level general-purpose programming language

Logo endorsed by the C++ standard committee

  - Created by Danish computer scientist Bjarne Stroustrup

Bjarne Stroustrup in his AT&T New Jersey office, c. 2000
[https://en.wikipedia.org/wiki/C%2B%2B]

  - Originally, it was an extension of C, i.e., "C with classes"

C++

- C++ was designed for system and embedded programming, resource-constrained software, and large systems
  - A light-weight abstraction programming language for building and using efficient and elegant abstractions [Stroustrup, 2015]
- Its design highlights are performance, efficiency, and flexibility
  - It has been found useful in many other contexts on resource-constrained applications
    - Desktop applications, video games, servers (e.g. e-commerce, web search, databases), performance-critical applications, etc.

C++

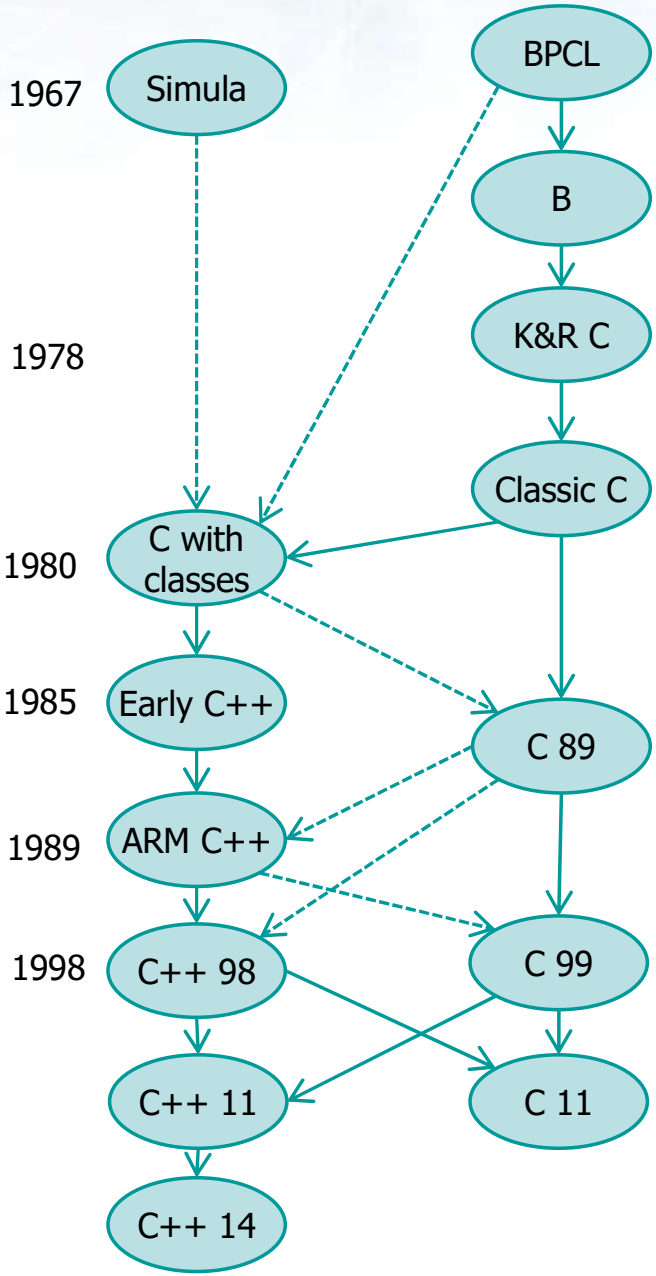
- The C++ language has two main components
  - A direct mapping of hardware features provided primarily by the **C subset**
  - A zero-overhead **abstractions** based on those mappings

C++

- The language has expanded significantly over time and modern C++ includes
  - Procedural — The program is composed by procedures
  - Imperative — Sequence of commands for the computer to execute
  - Generic programming — Algorithms are written in terms of programming to-be-specified later
  - Functional — Programs are constructed by composing functions
  - Object-oriented — Programs are organized around data (or objects)
  - Low-level construct — Programs allow low-level (e.g., memory) manipulation

History

- C++
  - Developed by Stroustrup at Bell Labs since 1979
    - As an extension of the C language "C with Classes" but including Simula and BCPL features
    - The target was to have an efficient and flexible language similar to C that also provided high-level features for program organization



History

- Initially standardized in 1998
  - International Organization for Standardization (ISO)
- Amended several times
  - Since 2012, C++ has been on a three-year release schedule with C++23 as the last planned standard

Informal Name	Standard	Comments
C++98	ISO/IEC 14882:1998	First Edition
C++03	ISO/IEC 14882:2003	Second Edition
C++11	ISO/IEC 14882:2011	Third Edition
C++14	ISO/IEC 14882:2014	Fourth Edition
C++17	ISO/IEC 14882:2017	Fifth Edition
C++20	ISO/IEC 14882:2020	Sixth Edition
C++23	ISO/IEC 14882:2020	December 2022

Standards

- C++11
  - Main features
    - Unified Initialization
    - Multithreading
    - Smart Pointers
    - Hash Tables
    - Container std::array
    - Move semantics
    - Lambda functions included
    - Added auto and decltype

Standards

- C++14
  - Main features
    - Generalized Lambdas
    - Reader-Writer Locks
    - Included constexpr
    - Return type deductions extended to all functions

Standards

- C++17
  - Main features
    - File system and network libraries
    - Improved Lambdas
    - Fold Expressions
    - Initializers in if and switch statements
    - Nested Namespaces
    - Transactional memory
    - Inline Variables
    - Optional header file
    - Concurrent and Parallel algorithms in Standard Template Library (STL)
    - Class Template argument deduction (CTAD)

Standards

- C++20
  - Supersedes all previous versions with new features and an enlarged standard library
  - Main features
    - Concepts library
    - 3-way comparisons
    - Map contains
    - Range-based for loop
    - New identifiers (import, module)
    - Calendar and time zone library
    - Functions std::string and std::to_array
    - Array bounded/unbounded
    - Likely and unlikely attributes