

B.3 — Introduction to C++17

 learncpp.com/cpp-tutorial/introduction-to-c17/

What is C++17?

In September of 2017, the [ISO \(International Organization for Standardization\)](#) approved a new version of C++, called C++17. C++17 contains a fair amount of new content

New improvements in C++17

For your interest, here's a list of the major changes that C++17 adds. Note that this list is not comprehensive, but rather intended to highlight some of the key changes of interest.

- `__has_include` preprocessor identifier to check if optional header files are available (no tutorial yet)
- if statements that resolve at compile time ([8.4 -- Constexpr if statements](#))
- Initializers in if statements and switch statements (no tutorial yet)
- inline variables ([7.9 -- Sharing global constants across multiple files \(using inline variables\)](#))
- Fold expressions (no tutorial yet)
- Mandatory copy elision for some cases (mentioned in [14.15 -- Class initialization and copy elision](#))
- Nested namespaces can now be defined as namespace `X::Y` ([7.2 -- User-defined namespaces and the scope resolution operator](#))
- Removal of `std::auto_ptr` and some other deprecated types
- `static_assert` no longer requires a diagnostic text message parameter ([9.6 -- Assert and static_assert](#))
- `std::any` (no tutorial yet)
- `std::byte` (no tutorial yet)
- `std::filesystem` (no tutorial yet)
- `std::optional` ([12.15 -- std::optional](#))
- `std::shared_ptr` can now manage C-style arrays (but `std::make_shared` can't create them yet) ([22.6 -- std::shared_ptr](#))
- `std::size` ([11.2 -- Arrays \(Part II\)](#))
- `std::string_view` ([5.10 -- Introduction to std::string_view](#))
- Structured binding declarations (no tutorial yet)
- Template deduction for constructors (no tutorial yet)
- Trigraphs have been removed
- `typename` can now be used (instead of `class`) in a template template parameter
- UTF-8 (u8) character literals (no tutorial yet)

[Next lesson](#)

[B.4Introduction to C++20](#)

[Back to table of contents](#)

[Previous lesson](#)

[B.2Introduction to C++14](#)