

Tutorial 0: Hello World

COMP6771 Advanced C++ Programming

Today's plan

1. Set up your environment
2. Write and compile your first program
3. Build with CMake
4. A taste of modern C++

Environment

You will need:

- **Compiler:** GCC 12+ or Clang 15+ (C++20 support)
- **Build system:** CMake 3.20+
- **Editor:** Your choice

On CSE machines, these are already available.

```
g++ --version
```

Hello World

```
#include <iostream>
#include <string>

auto main() -> int {
    auto const name = std::string{"COMP6771"};
    std::cout << "Hello, " << name << "!\n";
    return 0;
}
```

Course style conventions

- `auto main() -> int` -- trailing return type
- `auto const` for local variables where possible
- `std::string{"..."}` -- brace initialisation
- `\n` over `std::endl` -- avoids unnecessary flush

Compiling

```
g++ -std=c++20 -Wall -Wextra -Werror -o hello hello.cpp  
./hello
```

Flag	Purpose
<code>-std=c++20</code>	Enable C++20 standard
<code>-Wall -Wextra</code>	Turn on useful warnings
<code>-Werror</code>	Treat warnings as errors

Building with CMake

```
cmake_minimum_required(VERSION 3.20)
project(hello)

set(CMAKE_CXX_STANDARD 20)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

add_executable(hello hello.cpp)
```

```
cmake -B build
cmake --build build
./build/hello
```

A taste of modern C++

```
auto const numbers = std::vector{3, 1, 4, 1, 5, 9, 2, 6};

auto evens_doubled = numbers
    | std::views::filter([](int n) { return n % 2 == 0; })
    | std::views::transform([](int n) { return n * 2; });

for (auto const val : evens_doubled) {
    std::cout << val << " ";
}
// Output: 8 4 12
```

Don't worry if this looks unfamiliar -- it will be second nature by the end.

What's next

- Value semantics and the type system
- STL containers and iterators
- Move semantics and smart pointers
- Templates and generic programming
- Custom iterators and ranges

See you in Tutorial 1!