

# COMSW4995-6: Design using C++

Bjarne Stroustrup

## Informal description

Design cannot be understood in the abstract: To discuss design you need concrete examples – preferably examples of both good and bad design. Conversely, you cannot understand a programming language or library – or use it well – by just learning the rules for its individual features. You need to understand the general design ideas behind the language or library: Its philosophy. The ISO C++ language and its standard library provide many concrete examples for the discussion of design. We will look at C++ from its earliest days through the current 2020 ISO standard (C++20). This year’s version of this course will place some emphasis on the [C++ Core Guidelines](#) effort to provide tool-and-library supported guidelines for a modern style of C++ providing type and resource without loss of generality or performance.

This course involves a fair bit of reading, some programming, and some writing. Specific topics will be chosen from resource management (e.g., constructors and destructors), error handling (e.g., exceptions), generic programming (e.g. templates and concepts), compile-time computation, modularity, concurrency (threads and coroutines) and libraries (e.g. containers, algorithms, ranges, and smart pointers). Topics will be examined from various points of view, including usability, implementation models, teachability, performance, and real-world constraints.

## Course Description and Prerequisites

This course explores the interactions among language design, library design, and program design in the context of ISO standard C++. Features provided from early C++ to C++20 and the design and programming techniques they support are featured.

Requirements: Senior undergraduate, masters, professional, PhD graduate standing. A basic understanding of C++ and experience with a software development project (in any language) would be an advantage.

## Learning Outcomes or Course Objectives

After the course, the student will have a good knowledge of C++, an ability to evaluate language, library, and program designs. The student will also have some experience in putting this understanding into practical use as part of a design evaluation project, involving some programming, some writing (design evaluation and/or comparison and a tutorial).

## Teaching approach

Each week will feature a reading or programming assignment in support of the lecture. Attendance is essential.

## Textbook and/or Resource Material

Stroustrup: A Tour of C++ (Second edition) Addison Wesley 2018, the ISO C++ standard, [the C++ FAQ](#), papers on C++, [the Core C++ Guidelines](#), and various ISO C++ language extension and proposals.

## Grading

There will be a few (probably three) simple individual programming projects to solidify a student’s understanding of C++, a few reading assignments requiring written reports (in all 50%), and a major final 3-person team group project evaluating a library or language feature (40%). There will be a presentation of each final project. The deliverables for the final project will include design documentation and tutorial material. Class participation 10%.

## Time and place

Friday 10:10A-12:00P, Zoom.