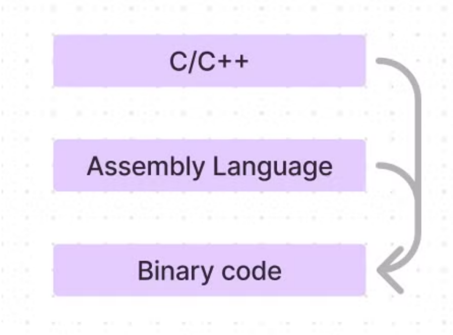
Section 1: Welcome

▼ 1.1 Welcome

▼ Overview

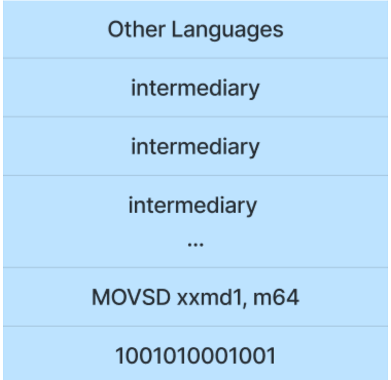
- C++ is a powerful programming language capable of low-level operations.
- It is a general-purpose language created by **Bjarne Stroustrup** as an extension of the C programming language.
- C++ is almost always implemented as a **compiled language**. Many vendors provide C++ compilers, including: Free Software Foundation, LLVM, Microsoft, Intel, Oracle, IBM
- Available on many platforms due to its wide compiler support.
- Features object-oriented, generic, and functional programming paradigms.
- C++ is close to the hardware like C but enables faster application development compared to C.

▼ Assembly Language:



- Assembly language is **hardware-specific** and non-portable.
- Example: Intel and ARM processors have their own unique assembly languages.

▼ Comparison with Other Languages:



Other Languages
intermediary
intermediary
intermediary ...
MOVSD xxmd1, m64
1001010001001

- Other languages like Python, Java, and JavaScript have many **intermediate layers** between them and the hardware.
- C/C++ is closer to the hardware, making it more efficient for low-level operations.
- C++ strikes the **sweet spot** between ease of implementation and being close to the hardware.

Applications

Bjarne Stroustrup's Homepage

This is the homepage of Bjarne Stroustrup, the designer and original implementor of C++

https://www.stroustrup.com/

High-Performance Applications:

- Games, Financial applications, Highly responsive GUI applications, Servers and database systems, Network infrastructure applications, Space applications, Manufacturing systems, Medical devices

Companies Using C++:

- Adobe, Alias/Wavefront Maya, Amazon, Apple (macOS), Autodesk, CERN, Facebook, and many others

About the Course

How This Course is Structured

- Prepares you for **beginner to intermediate job roles** in the market
- Short, **code-based slides**
- **Code examples** in IDEs
- Built-in **practice**
- **Quizzes and exercises**

This Course Covers Modern C++

- **Classic C++**: C++ 98 and earlier
- **Modern C++**: C++ 11 and later
    - This course focuses on **modern C++**: C++ 11, C++ 14, C++ 17, and C++ 20

Key Features of C++ 20

- C++ 20 introduces 4 major features:
    - **Ranges**
    - **Modules**
    - **Coroutines**
    - **Concepts**

- Additionally, you will learn about other modern concepts like:
    - **Lambdas**
    - **Move semantics**
    - **Smart pointers**

▼ 1.2 Getting the most out of the course

- This course provides tutorials.
- Serves as a reference.
- This course will teach you to use the documentation.