

Lecture 1:

Welcome to CS106L!

CS106L, Fall 2026

Preston Seay & Emily Ai

Today's Agenda

- Introductions!
- The Pitch
- Course Logistics

Introductions

Preston Seay

- Junior
- CS Systems
- Education
- Vibe Coding
- Outdoors
- Cyber



Emily Ai

- Sophomore
- CS + Data Science
- cat lover <3
- love horror/mystery movies & games!



RESIDENT EVIL
DEAD SPACE
INSCRIPTION



Now you can meet (some of) each other!

- Turn to the people next to you and introduce yourselves!
- **Potential Conversation Topics:**
 - What are some of your hobbies?
 - Why do you want to take this class?
 - ***Would you rather object at every wedding you go to or accidentally laugh out loud at every lecture you attend?***



The Pitch

Why C++?

“The invisible foundation of everything”

Valorant



[source]

CS:GO/CS2



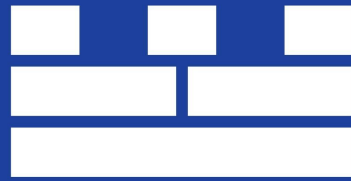
[source]

...and many more!



High Frequency Trading

Optiver 



CITADEL | Securities



[source]

Self Driving



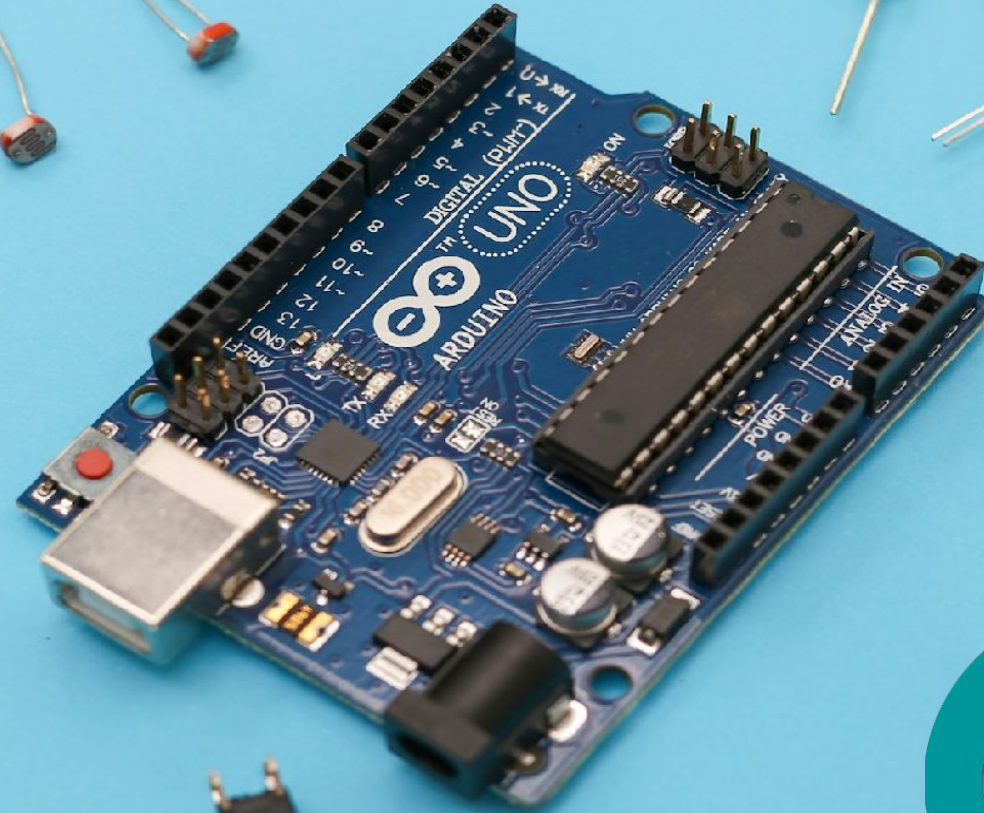
TESLA



WAYMO

[source]

Arduino



[source]

GPU Programming



NVIDIA®

CUDA®

And much, much more!

- Databases (MySQL, MongoDB)
- Web Browsers (Chrome, Safari, Edge)
- Virtual Reality (Quest)
- Low level ML (PyTorch, TensorFlow)
- Compilers, virtual machines (JVM, LLVM, GCC)
- Operating Systems (Windows, MacOS, Linux)

“The invisible foundation of everything”

C++ is great for...

- Handling lots of data
- And handling it very efficiently
- And doing it in an elegant, readable way

C++ in Industry

The Amazon logo, featuring the text "amazon.com" in a black sans-serif font with a registered trademark symbol, and a curved orange arrow underneath the text pointing from the 'a' to the 'm'.The Meta logo, consisting of a blue infinity symbol followed by the word "Meta" in a dark blue sans-serif font.The Google logo, featuring the word "Google" in its multi-colored sans-serif font: blue 'G', red 'o', yellow 'o', blue 'g', green 'l', and red 'e'.

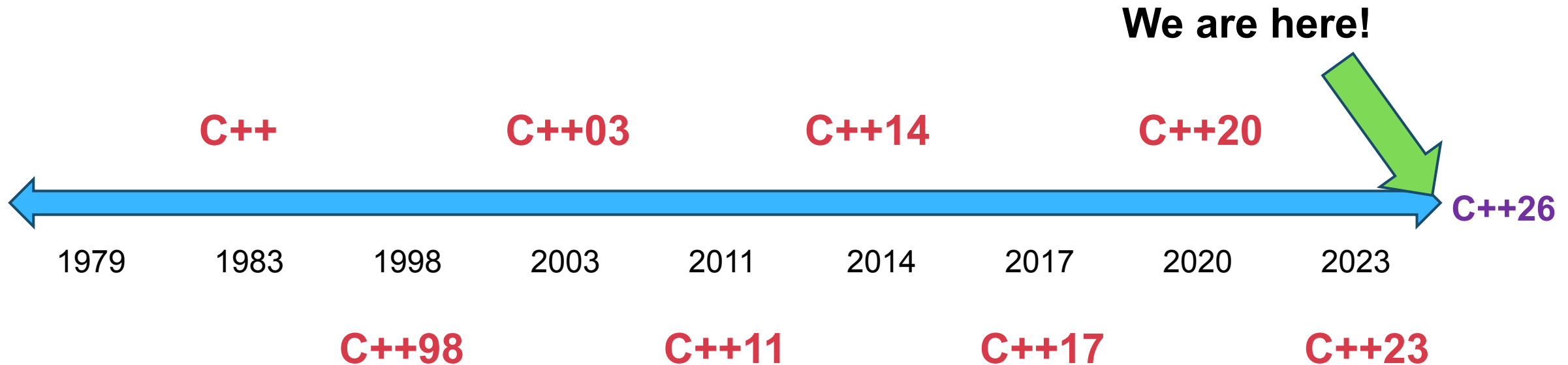
C++ was first released in 1985, still #3!

Mar 2026	Mar 2025	Change	Programming Language		Ratings	Change
1	1			Python	21.25%	-2.59%
2	4	▲		C	11.55%	+2.02%
3	2	▼		C++	8.18%	-2.90%
4	3	▼		Java	7.99%	-2.37%
5	5			C#	6.36%	+1.49%
6	6			JavaScript	3.45%	-0.01%
7	9	▲		Visual Basic	2.50%	-0.02%
8	8			SQL	2.00%	-0.57%
9	16	▲▲		R	1.88%	+0.94%
10	10			Delphi/Object Pascal	1.80%	-0.36%

[TIOBE Index, March 2026]

The C++ Community

- C++ has a **MASSIVE** user base
- C++ Standard continues to be revised every three years
 - A standards committee decides new changes to the language



What is C++?

A valid C++ program

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

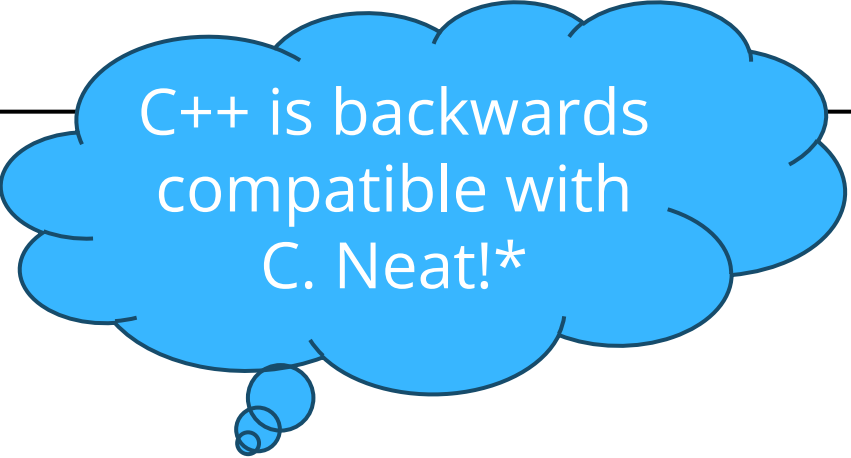
int main() {
    auto str = std::make_unique<std::string>("Hello World!");
    std::cout << *str << std::endl;
    return 0;
}

// Prints "Hello World!"
```

Also a valid C++ program

```
#include "stdio.h"
#include "stdlib.h"

int main(int argc, char *argv) {
    printf("%s", "Hello, world!\n");
    // ^a C function!
    return EXIT_SUCCESS;
}
```



C++ is backwards
compatible with
C. Neat!*

Also a valid C++ program

```
#include "stdio.h"
#include "stdlib.h"

int main(int argc, char *argv) {
    asm(".LC0:\n\t"
        ".string \"Hello, world!\"\n\t"
        "main:\n\t"
        "push rbp\n\t"
        "mov rbp, rsp\n\t"
        "sub rsp, 16\n\t"
        "mov DWORD PTR [rbp-4], edi\n\t"
        "mov QWORD PTR [rbp-16], rsi\n\t"
        "mov edi, OFFSET FLAT:.LC0\n\t"
        "call puts\n\t");
    return EXIT_SUCCESS;
}
```



C++ History: Assembly

```
section .text
global _start                ;must be declared for linker (ld)
_start:                      ;tell linker entry point
    mov edx,len              ;message length
    mov ecx,msg              ;message to write
    mov ebx, 1               ;file descriptor (stdout)
    mov eax, 4               ;system call number (sys_write)
    int 0x80                 ;call kernel
    mov eax, 1               ;system call number (sys_exit)
    int 0x80                 ;call kernel
section .data
    msg db 'Hello, world!' ,0xa ;our dear string
    len equ $ - msg           ;length of our dear string
```

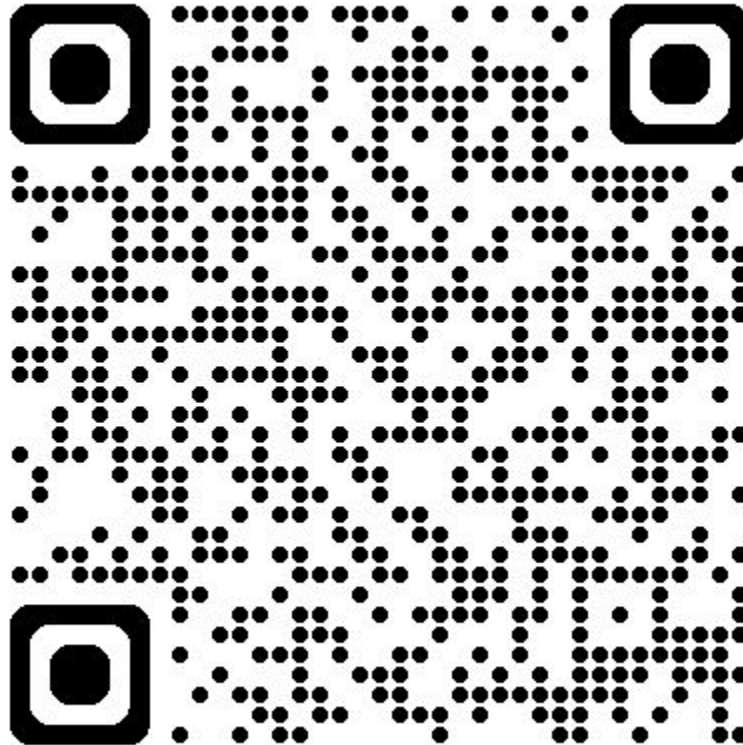
C++ History: Assembly

-  Unbelievably **simple** instructions
-  Extremely **fast** (when well-written)
-  Complete **control** over your program

Why don't we always use assembly?

Live demo - exploring compilers!

<https://godbolt.org/z/TEqs6Ts1x>



C++ History: Assembly

-  Unbelievably **simple** instructions
-  Extremely **fast** (when well-written)
-  Complete **control** over your program

-  A **lot of code** (even for simple tasks)
-  Very **hard to understand**
-  Extremely **unportable**

C++ History: Invention of C

- Dennis Ritchie created C in 1972 to much praise.
- C made it easy to write code that was:
 - Fast
 - Simple
 - Cross platform
 - **Compilers!** Source Code → Assembly
- Learn to love it in **CS107!**



Dennis Ritchie (left), alongside Ken Thompson (right)

C++ History: Invention of C

- C was popular because it was simple
 - *“When I read C I know what the output Assembly is going to look like”*
—Linus Torvalds, creator of Linux
- However, C has some weaknesses:
 - No **objects** or classes
 - Difficult to write **generic or templated** code
 - **Tedious** to write large programs

C++ History: Welcome to C++!

- In 1983, the beginnings of C++ were created by Danish computer scientist Bjarne Stroustrup



Bjarne Stroustrup, the man himself 😊

C++ Design Philosophy

- Express ideas and intent directly in code.
- Enforce safety at compile time whenever possible.
- Do not waste time or space.
- Compartmentalize messy constructs.
- Allow the programmer full control, responsibility, and choice.

"Code should be elegant **and** efficient; I hate to have
to choose between those"

—Bjarne Stroustrup

C++ Design Philosophy (Summarized)

- Readable
- Safety
- Efficiency
- Abstraction
- Programmer Choice

Why take CS106L?

Topics We'll Cover

Welcome	Types & Structs
Initialization & References	Streams
Containers	Iterators & Pointers
Classes	Advanced Classes
Templates	Advanced Templates
Functions & Lambdas	Operator Overloading
Special Member Functions	Move Semantics
std::optional and Type Safety	RAII, Smart Pointers, C++ Projects

Topics We'll Cover

Welcome	Types & Structs
Initialization & References	Streams
Containers	Iterators & Pointers
Classes	Advanced Classes
Templates	Advanced Templates
Functions & Lambdas	Operator Overloading
Special Member Functions	Move Semantics
std::optional and Type Safety	RAII, Smart Pointers, C++ Projects

Other Classes (e.g. CS106B)

vs.

CS106L

- Focus on **concepts** like abstractions, recursion, pointers etc.
- **Bare minimum** C++ in order to use these concepts
- **Heavy** time commitment

- Focus is on **code**: what makes it good, what powerful and elegant code looks like
- The real deal: No Stanford libraries, only STL: understand **how** and **why** C++ was made
- Relaxed **1 unit** class!

You will probably use C++...

- In at least one of Stanford's classes
 - **CS 111:** Operating Systems Principles
 - **CME 213:** Introduction to parallel computing using MPI, openMP, and CUDA
 - **CS 143:** Compilers
 - **CS 144:** Introduction to Computer Networking
 - **CS 248A:** Computer Graphics: Rendering, Geometry, and Image Manipulation
 - **MUSIC 256A:** Music, Computing, Design: The Art of Design
 - ...and more!
- And in real life!

C++ helps develop good coding hygiene

- Am I using objects the way they're meant to be used?
 - Type checking, type safety
- Am I using memory efficiently?
 - Reference/copy semantics, move semantics
- Am I modifying something I'm not supposed to?
 - **const** and const correctness
- Other languages relax these restrictions



“Nobody should call themselves a professional if they only know one language” —Bjarne Stroustrup

A pixelated Creeper from the game Minecraft stands in the center of a dirt pit. The Creeper is green with a black face and is looking directly at the viewer. The pit is made of brown dirt blocks, and the ground around the Creeper is covered in green grass blocks. The lighting is dim, suggesting an underground or nighttime setting.

What questions do we have?

creeper_abt_to_explode_with_questions

Course Logistics

Asking Questions

- Please ask questions!
- Feel free to raise your hand at **any time** with a question
- We'll also pause periodically to solicit questions and check understanding



Access and Accommodations

- Students with disabilities are a valued and essential part of the Stanford community. We welcome you to our class!
- Please work with OAE but also let us know if there's anything we can do to make the course more accessible to you.
- Don't be shy about asking for accommodations if problems arise. We will do whatever we can to help.

Community Norms

- Shame-free zone
- Treat your peers and instructors with kindness and respect
- Be curious
- Communication is key!

Guiding Principles

- We will do everything we can to support you. We want to provide flexibility to the best of our ability!
- We want to hear your feedback so we can ensure the class is going as smoothly as possible for everyone
- Please communicate with us if any personal circumstances or issues arise! We are here to support you :)

What questions do you have?



bjarne_about_to_raise_hand

Lecture

- Held **Tuesdays and Thursdays** from **3:00PM – 4:20PM** in **Thornton 110**
- Lecture is recorded
- **Attendance is required.** To receive attendance credit, you must **fill out attendance codes at the beginning of class.** These will only be available for the **first 10 minutes of each class.**
- If you're absent, you'll need to complete an additional assignment on top of your class activity for the day

Illness/Emergencies

- If you are sick, for the wellbeing of yourself and others **please stay home**, take care of yourself, and **reach out to us** – **we never want** you to feel that you must attend class if you are not feeling well!
- Similarly, if you have an emergency or exceptional circumstance, **please reach out to us** so that we can help!

Office Hours

- OH times are TBD and will be in person
 - Will be settled by week 2
 - Will be posted on Ed
- We want to meet you! Come talk!
- Watch the [course website](#) and [Ed](#) for more info.

Assignments

- There will be a short in class activity to complete each day!
 - They should be very do-able if you follow along during lecture
 - Available at 106l.com
 - You can enter “hello world” for the attendance code today
- We will have two larger assignments during the quarter that counts as a “midterm” and “final”

Grading

- Grading is S/NC. We expect everyone to get an S!
- How do you get an S?
 - Attend **all** lectures between Week 2 and Week 9
 - Complete the class activity for each lecture!
 - Successful completion of **2 assignments** (midterm and final)

Get in touch with us!

- Here are the best ways to communicate with us!
- Email us: cs106l-aut2627-staff@lists.stanford.edu
 - Please use this email and not our individual emails so we both receive the message!
- Public or private post on Ed
 - Join the Ed via Canvas
- After class or in our office hours

A Creeper from the game Minecraft is standing in a dirt pit. The Creeper is green with a pixelated texture and a black cross-shaped mouth. It is surrounded by brown dirt blocks and green grass blocks. The text "see you all on thursday :)" is overlaid in white.

see you all on thursday :)

creeper_abt_to_explode_with_EXCITEMENT