

**SPRING 2026**  
**CS 103000**

**Professor**  
**Madeline Blount**  
**she/her**

**Week 0**

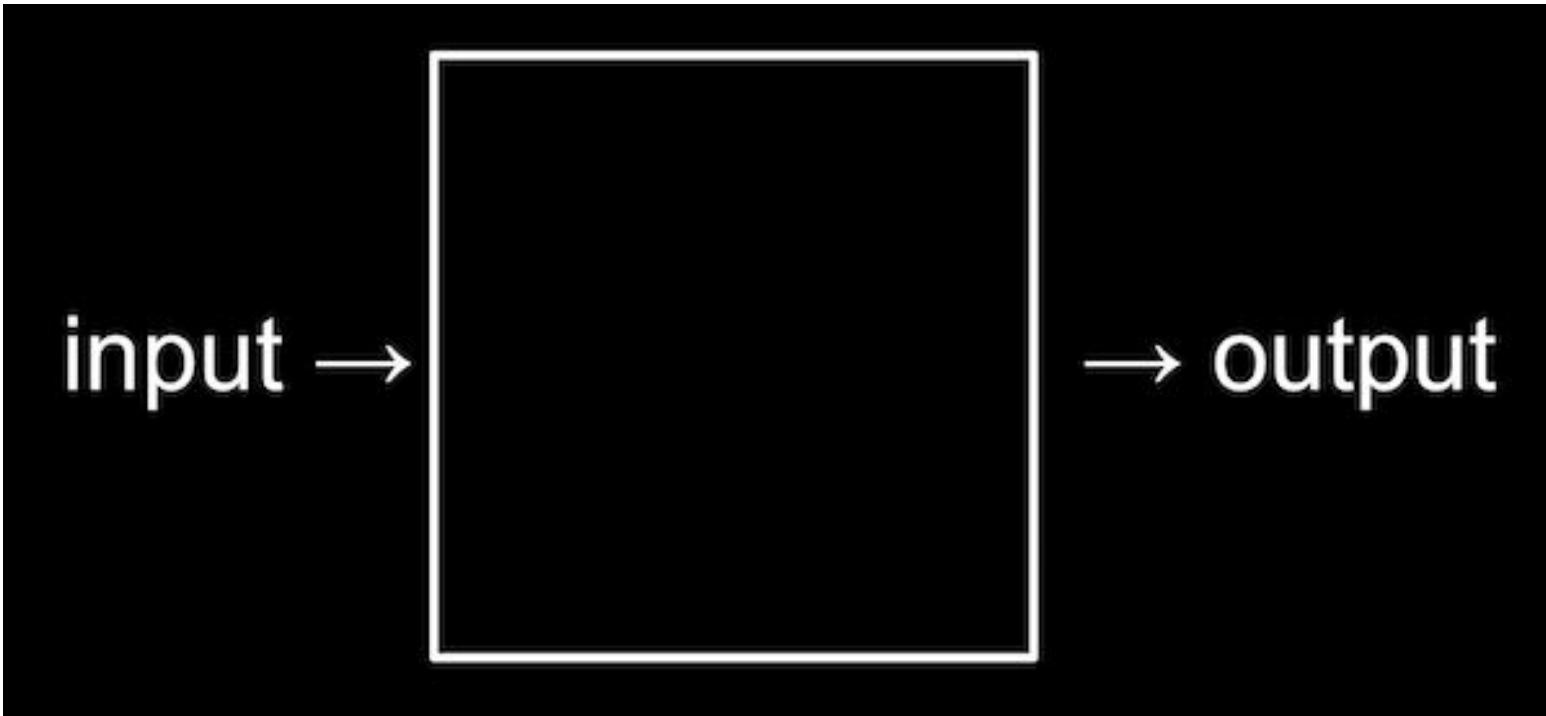


*Dall-E 2: cats learning C++ in the forest on '90's technology*

**Professor  
Madeline Blount  
she/her**



# COMPUTATIONAL THINKING:



Human language:

I want it to say "hello  
world"

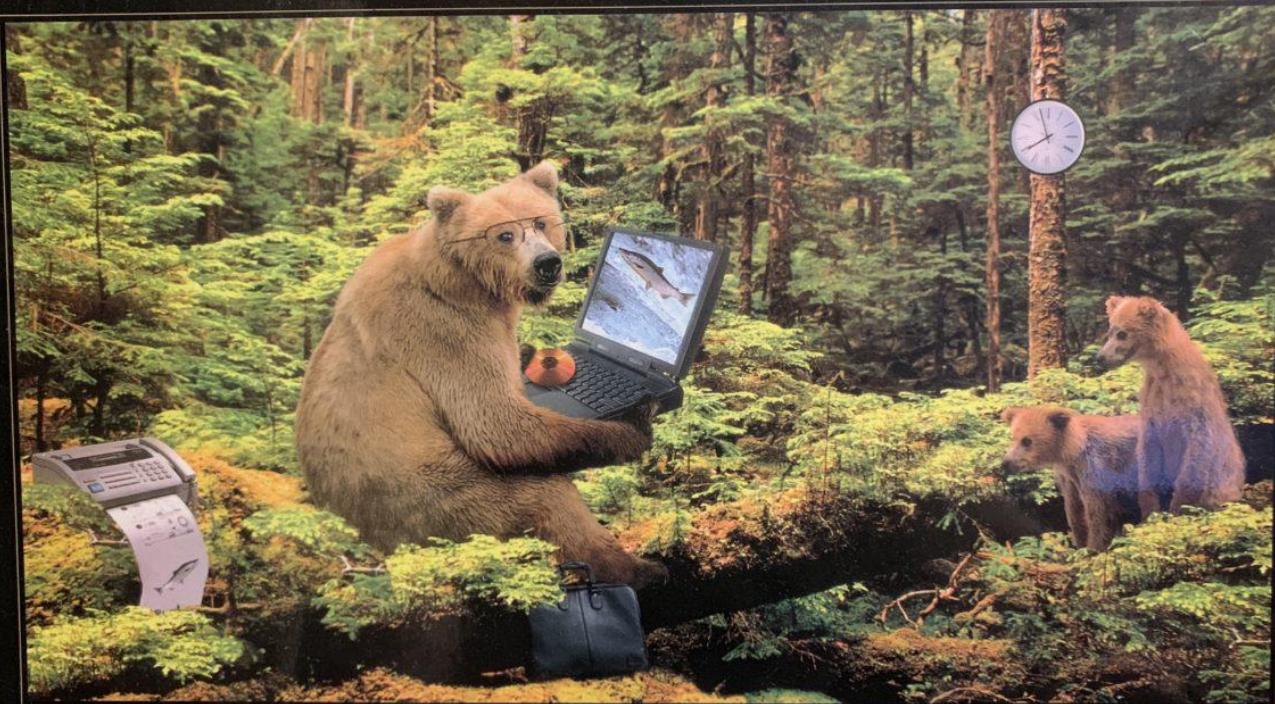
High-level languages: C++

```
cout << "hello world";
```

machine instructions: BINARY

```
010101110100101001001010100  
101000101001010010101001010
```

# L·E·A·R·N·I·N·G



LEARNING IS DISCOVERING THAT NOTHING IS IMPOSSIBLE.

ENGLISH:

How are you?

GREEK:

τι κάνεις;

C++:

```
cout << "Hello World!" << endl;  
cout << "How are you?" << endl;
```

```
Hello World!  
How are you?
```

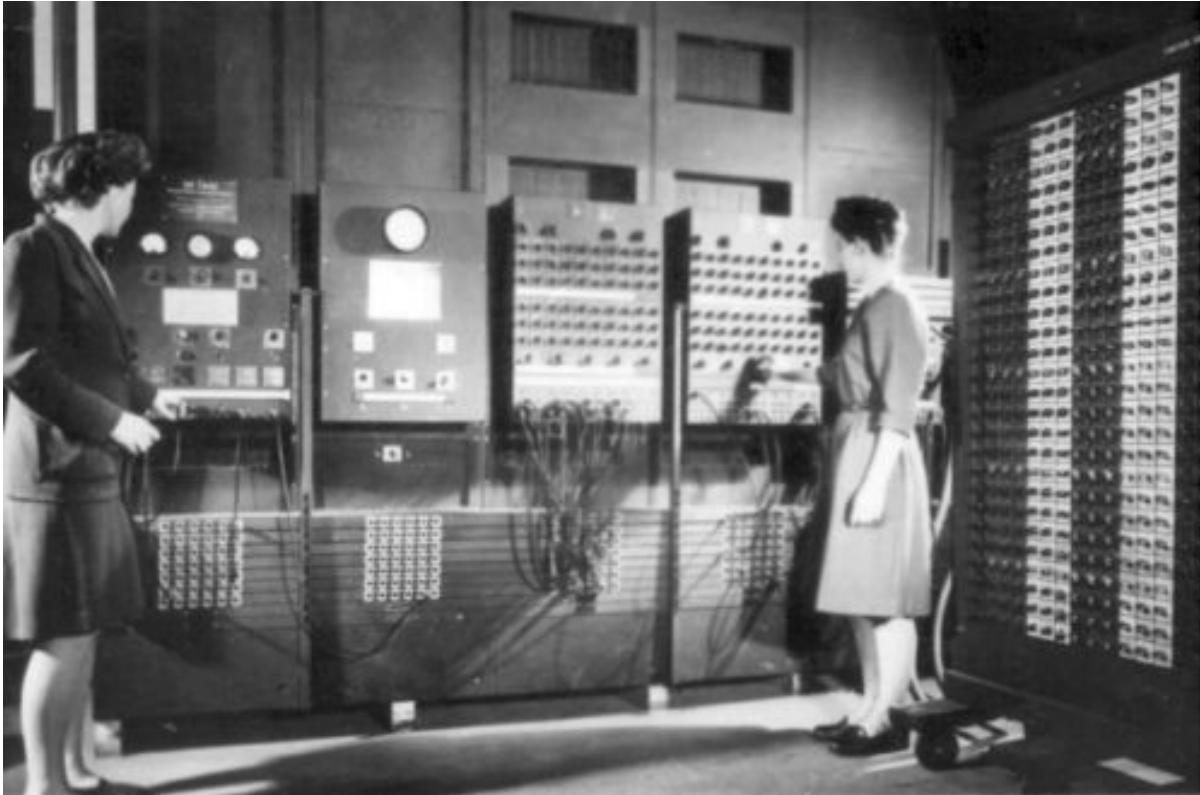
# BINARY (machine instructions):

01001000  
00100000  
01100100



(Hello World!)

early computers, 1930's-1940's



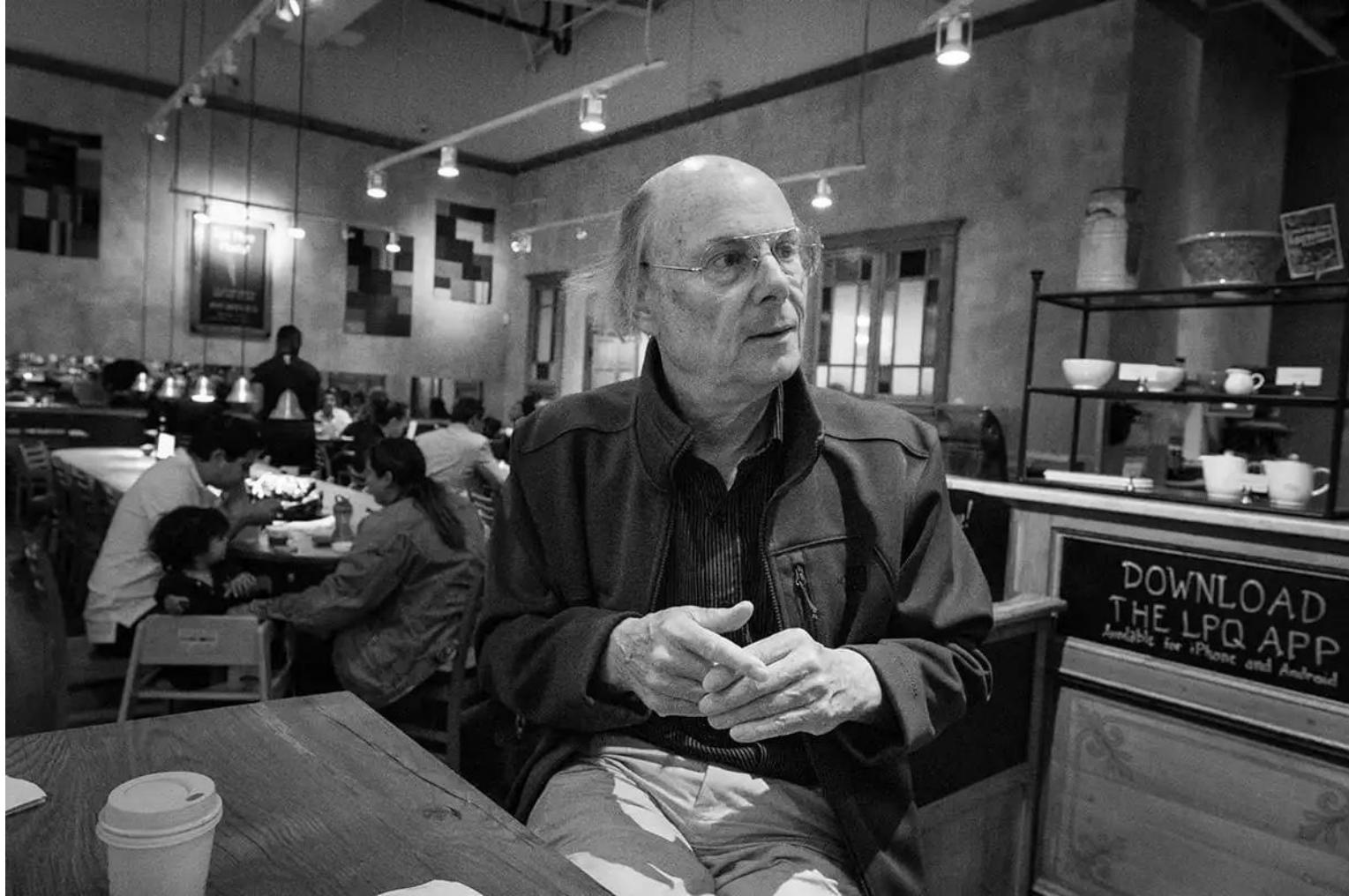
zyBooks, from Army Archives

# Why C++?

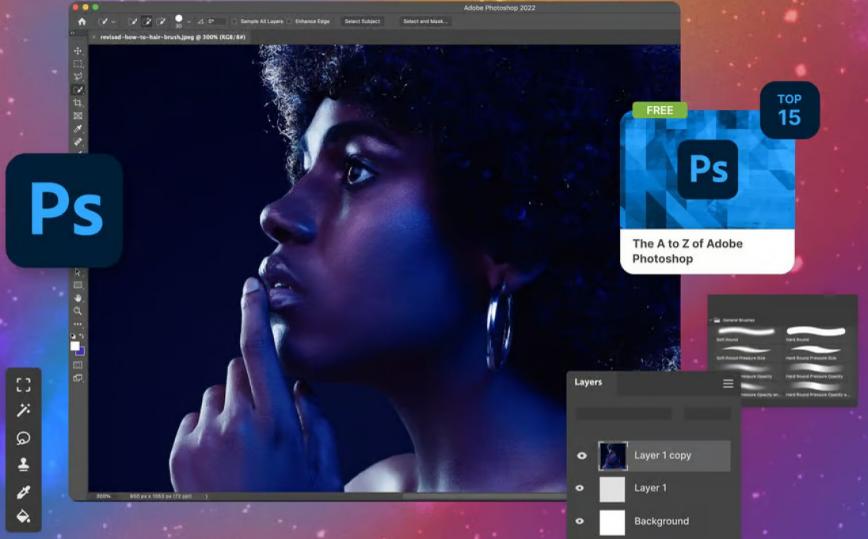
- Had to choose some language!!
- Good foundational language
- Fast, efficient w/memory
- *What we learn here can translate to other languages.*

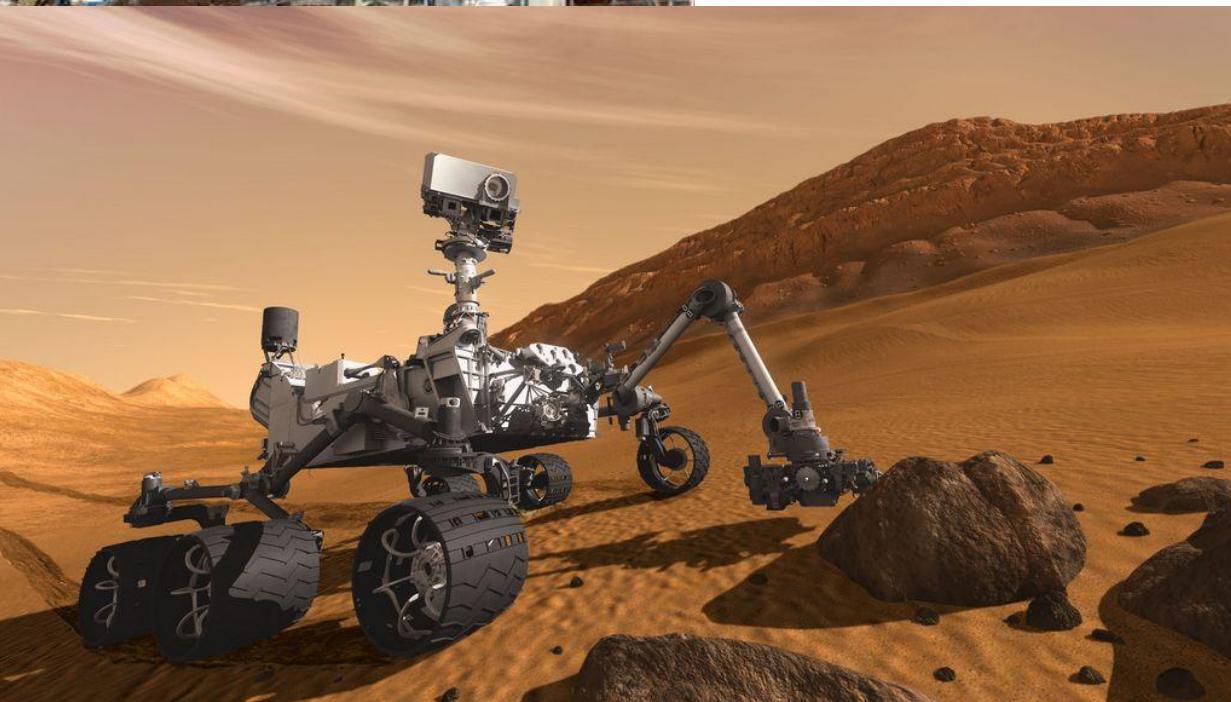
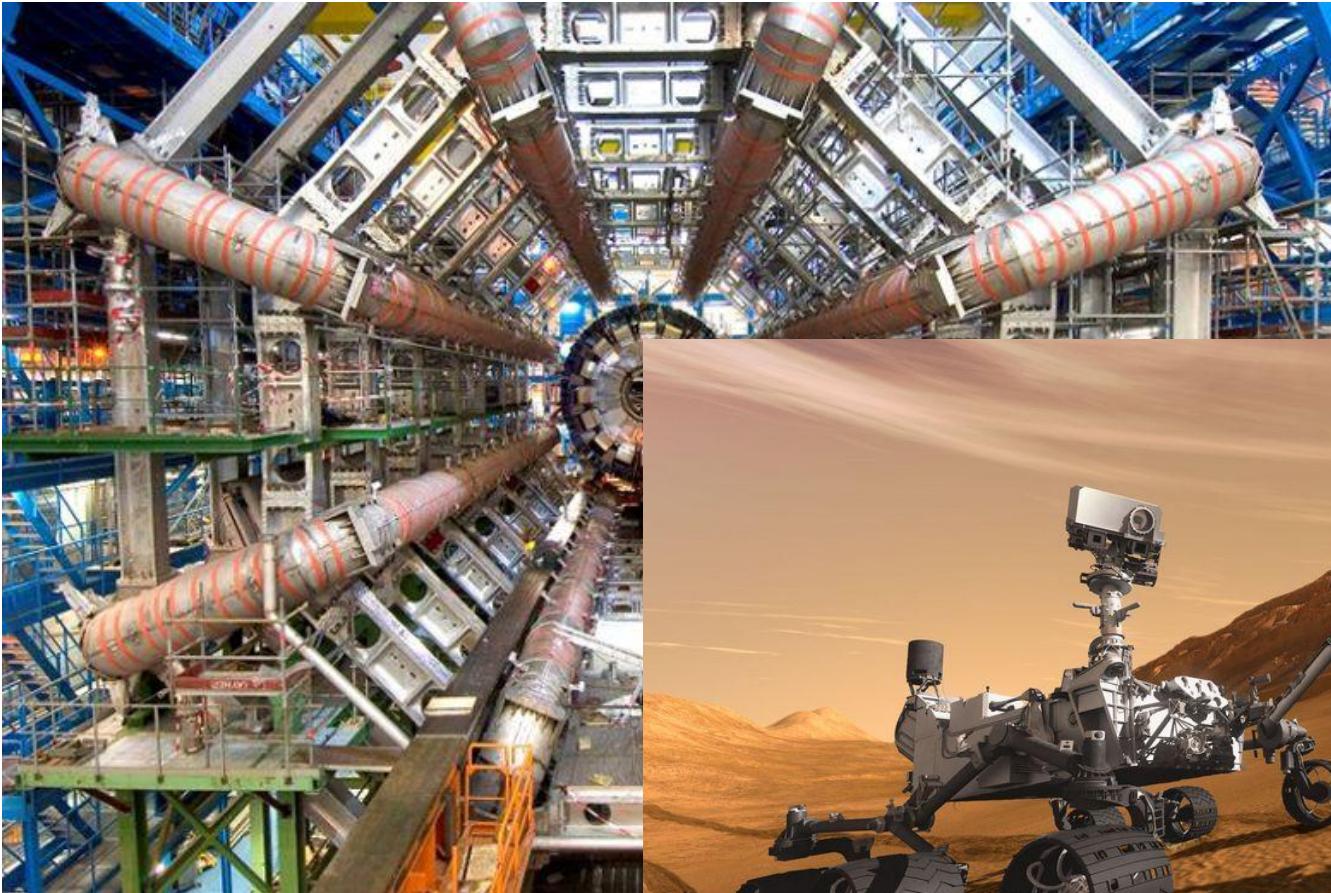
## Bjarne Stroustrup:

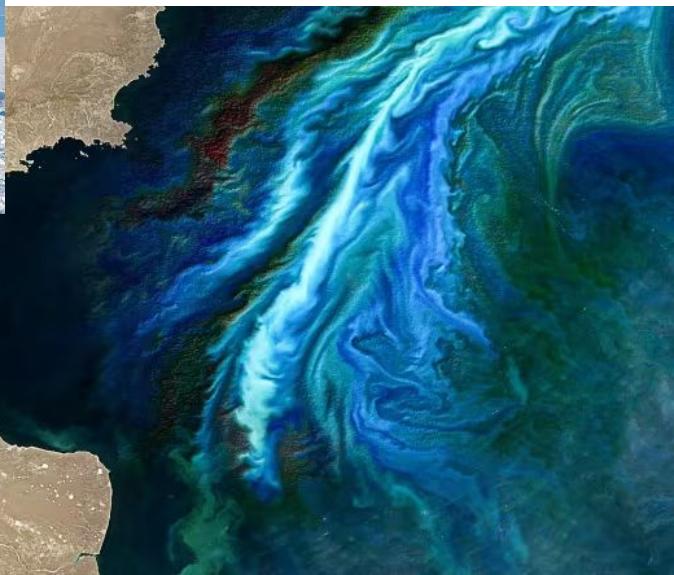
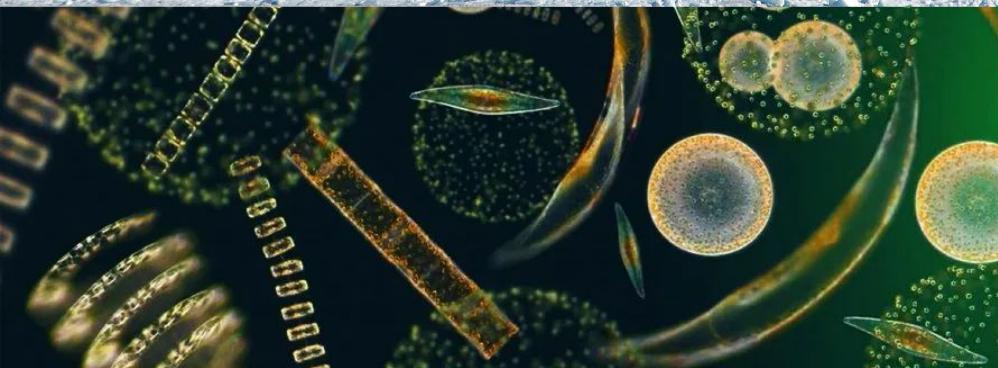
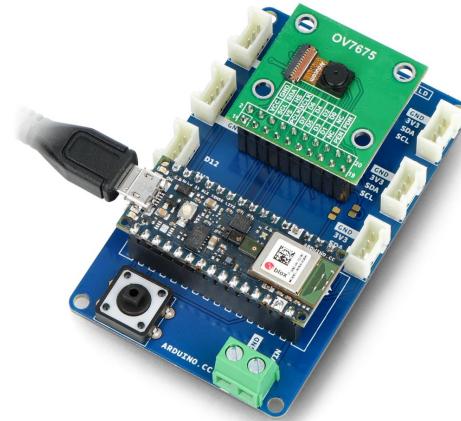
"The aims of C++ are noble: enable programmers to write real-world programs that are simultaneously elegant and efficient, to raise the level of abstraction in real-world code, and thereby improve the working lives of hundreds of thousands of serious programmers."

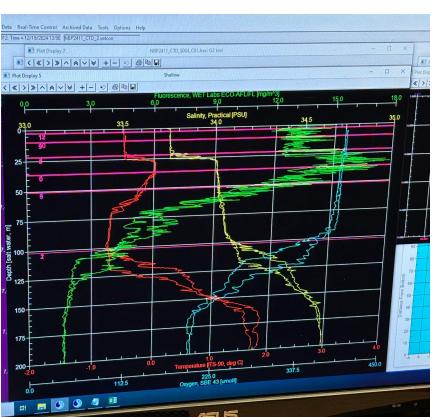


Codecademy, 2019









National  
Science  
Foundation



## Grace Hopper:

“To me programming is more than an important practical art. It is also a gigantic undertaking in the foundations of knowledge.”



## McKenzie Wark, A Hacker Manifesto:

“Hackers create the possibility of new things entering the world.”

## Ursula Franklin (1989 lecture):

"As I see it, technology has built the house in which we all live. The house is continually being extended and remodelled. More and more of human life takes place within its walls, so that today there is hardly any human activity that does not occur within this house. All are affected by the design of the house, by the division of its space, by the location of its doors and walls. Compared to people in earlier times, we rarely have a chance to live outside this house. And the house is still changing ..."'

MON.

TUES.

WED.

THRS.

FRI.

lecture

A little  
more  
reading ...

lecture

READING DUE

More work  
on labs ...

Lab section

LABS DUE

Start some  
reading ...

## How to do well in this class?

- Come to **class** - usually, without devices
- Be **patient** and **curious** about the **process** of programming
- ⏳ Give yourself enough **TIME!**



"I was taught ... that one ought to figure out a program completely on paper before even going near a computer. I found that I did not program this way ... instead of patiently writing out a complete program and assuring myself it was correct, I tended to just spew out code that was hopelessly broken, and gradually beat it into shape. Debugging, I was taught, was a kind of final pass where you caught typos and oversights. The way I worked, it seemed like **programming consisted of debugging**.

If I had only looked over at the other makers, the painters or the architects, I would have realized that there was a name for what I was doing: sketching ... **You should figure out programs as you're writing them, just as writers and painters and architects do.**"

-Paul Graham, software engineer + writer

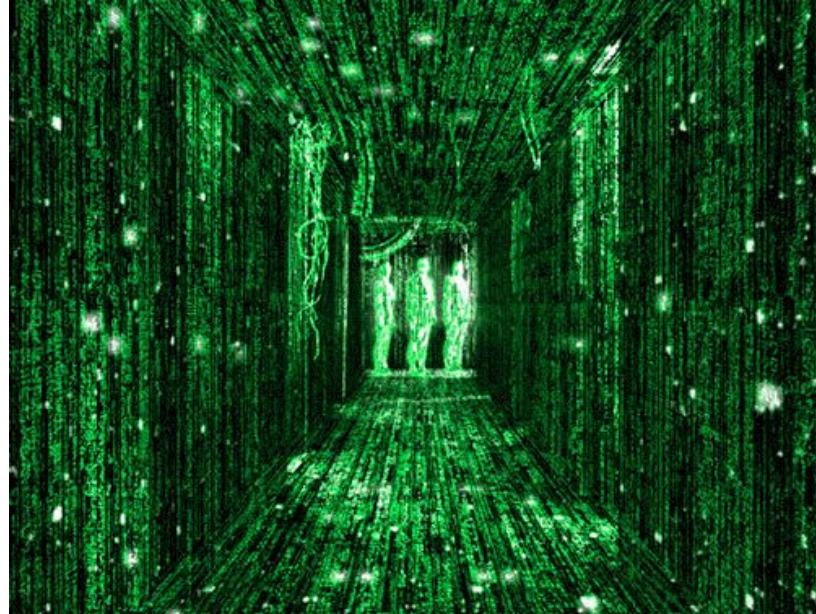


**1080p**  
Upscaled to  
**4K**





**“I know kung fu”**



```
1 #include <iostream>
2 using namespace std;
3
4 ▼ int main( ) {
5     cout << "Hello World!" << endl;
6 }
```

## Week 0: Assignment

- **Read syllabus!**
- Instructions for this first set of tasks are in there
- Get your zyBooks! Start Week 1
- Log into Discord - invite soon
- Read class survey
- 🧢 GO TO LAB! Fri. Jan. 30th
- **SURVEY DUE: Friday Jan. 30th @ 11:59pm**



SEE YOU NEXT LECTURE: Wed. Jan. 28th