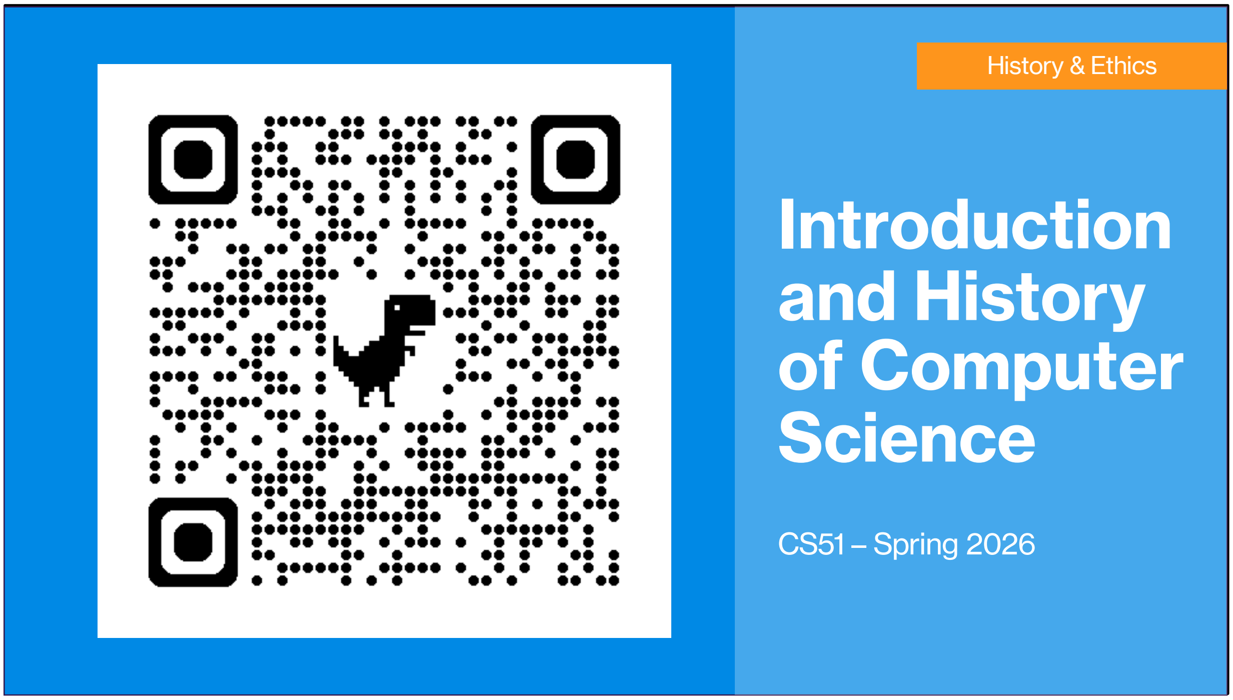


History & Ethics

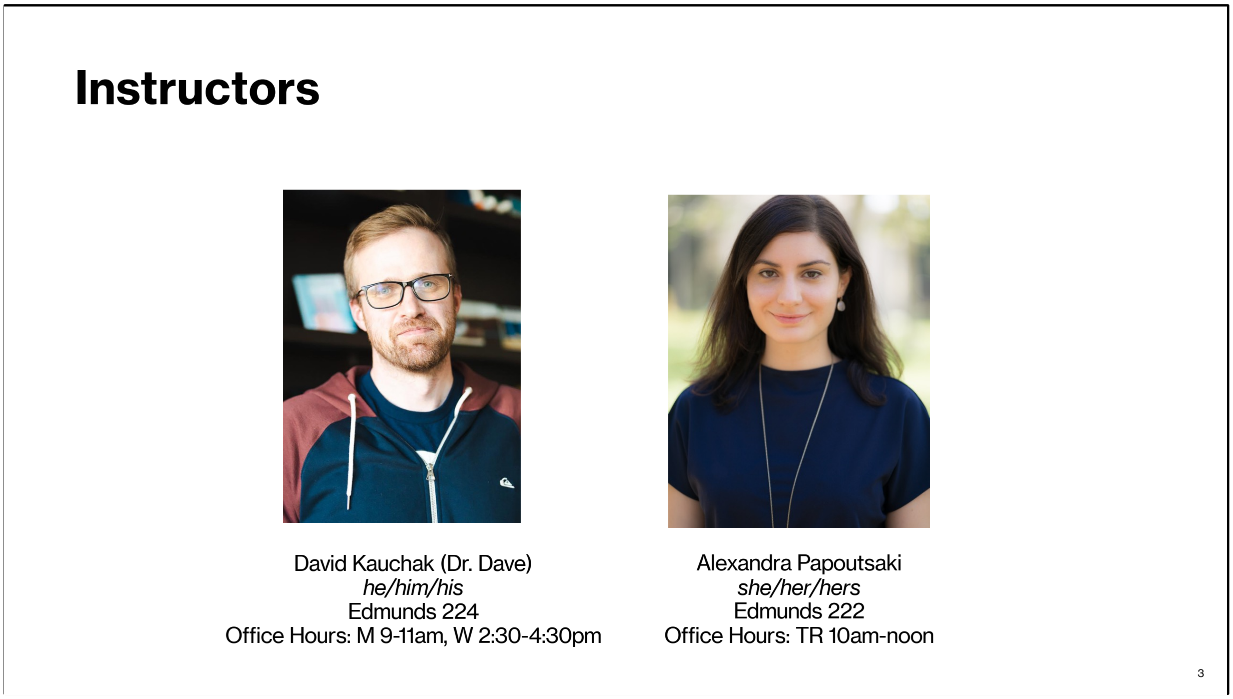
Introduction and History of Computer Science

CS51 – Spring 2026

Before we start, here is a link to the course website where you can find the schedule, syllabus, and the lecture slides posted if you want to follow along.

Welcome!

Welcome to CSCI051 or CS51 for short.



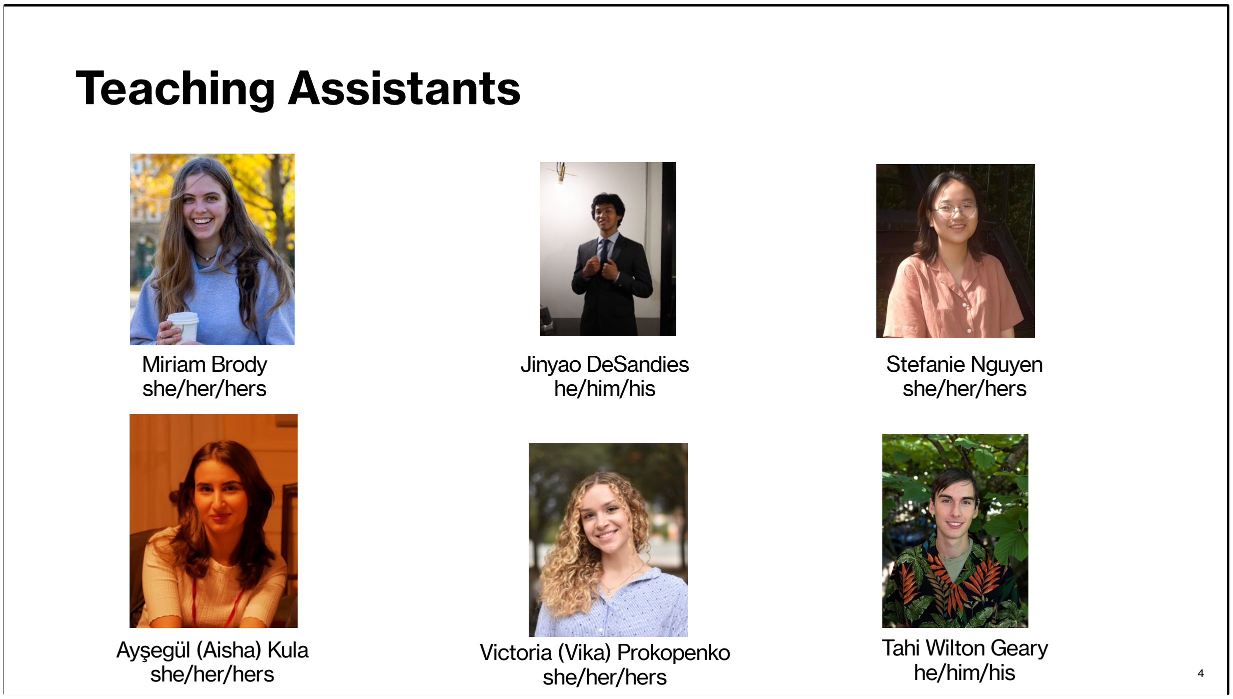
Instructors

David Kauchak (Dr. Dave)
he/him/his
Edmunds 224
Office Hours: M 9-11am, W 2:30-4:30pm

Alexandra Papoutsaki
she/her/hers
Edmunds 222
Office Hours: TR 10am-noon

3

The two instructors for the class are Prof Kauchak, who goes by Dr. Dave, and Professor Papoutsaki (Pa-poo-TSA-kee). Together, we will be covering the Tuesday/Thursday lectures and Friday labs. If you have been registered for the class, you need to also register for one of the two Friday lab for which attendance is mandatory. Here are our office hours for the semester although they might change so always keep an eye for announcements. We want you to come to our office hours with questions about the course material, assignments, College, post-graduate advice, and anything else that is going on.



Teaching Assistants

Miriam Brody
she/her/hers

Jinyao DeSandies
he/him/his

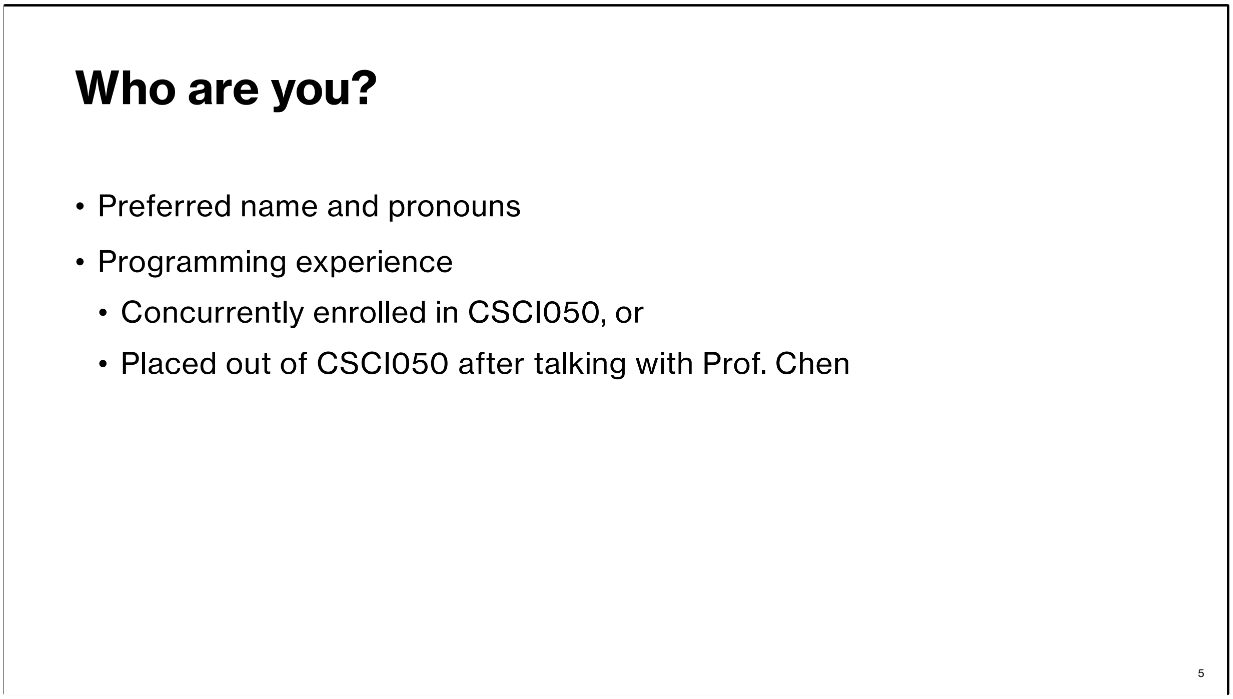
Stefanie Nguyen
she/her/hers

Ayşegül (Aisha) Kula
she/her/hers

Victoria (Vika) Prokopenko
she/her/hers

Tahi Wilton Geary
he/him/his

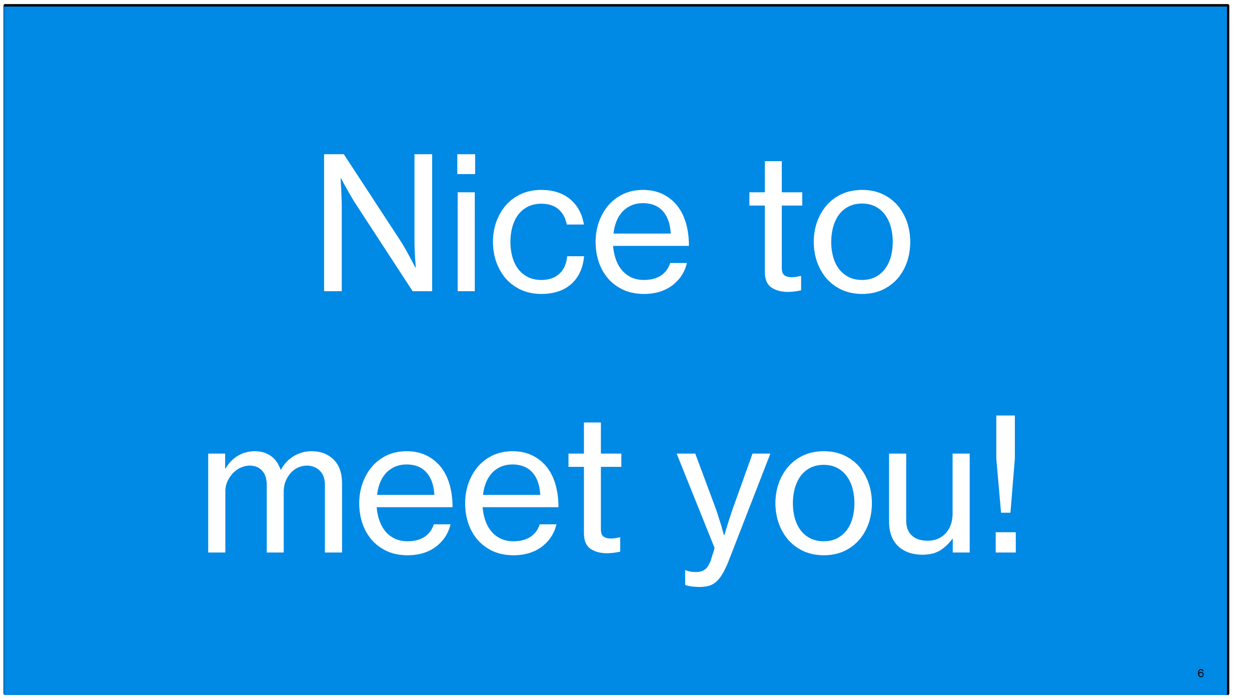
We also have an incredible team of TAs that will be assisting with the grading of the assignments, the Friday labs, and mentor sessions. TAs are one of the most valuable resources to the learning environment our department offers. Mentor sessions are a place that you can go not only for help with assignments but also to meet other classmates and hear advice from more experienced peers. Half of our TAs have Taed CS51 before and half took it last semester so we have a good mix of people that know how to help you and can really empathize with you. The schedule for mentor sessions will be announced at the end of this week.



Who are you?

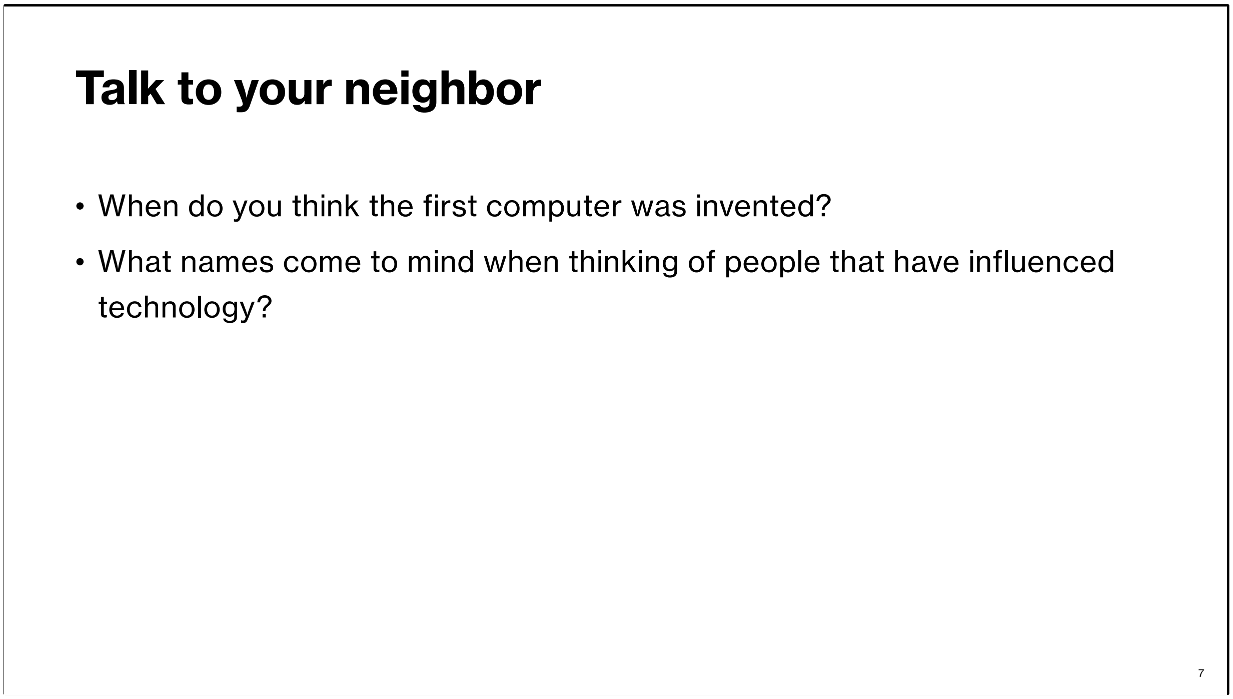
- Preferred name and pronouns
- Programming experience
  - Concurrently enrolled in CSCI050, or
  - Placed out of CSCI050 after talking with Prof. Chen

Now I would like us all to get a better sense of who’s in the room. Can you please state your name and pronouns, if you prefer, and whether you are currently enrolled in CSCI050 or if you have placed out of it and if yes what is your programming experience?



Nice to meet you!

Nice to meet you all! We are very excited to partner with you through this learning journey

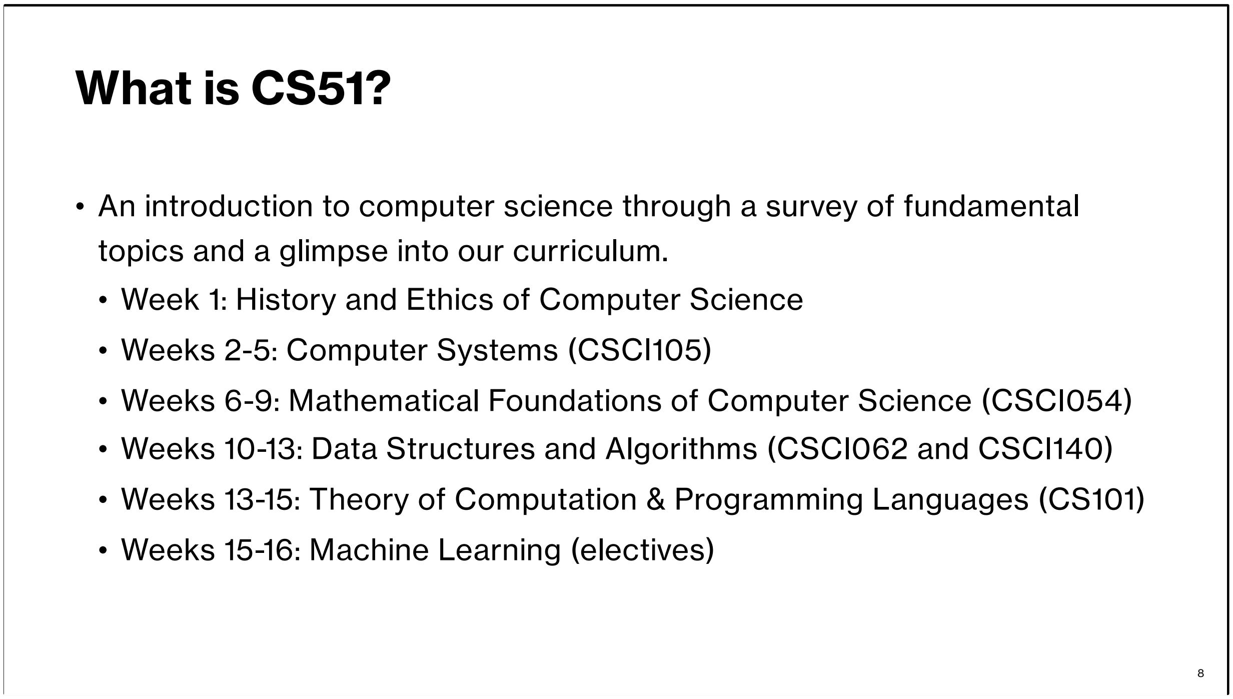


Talk to your neighbor

- When do you think the first computer was invented?
- What names come to mind when thinking of people that have influenced technology?

7

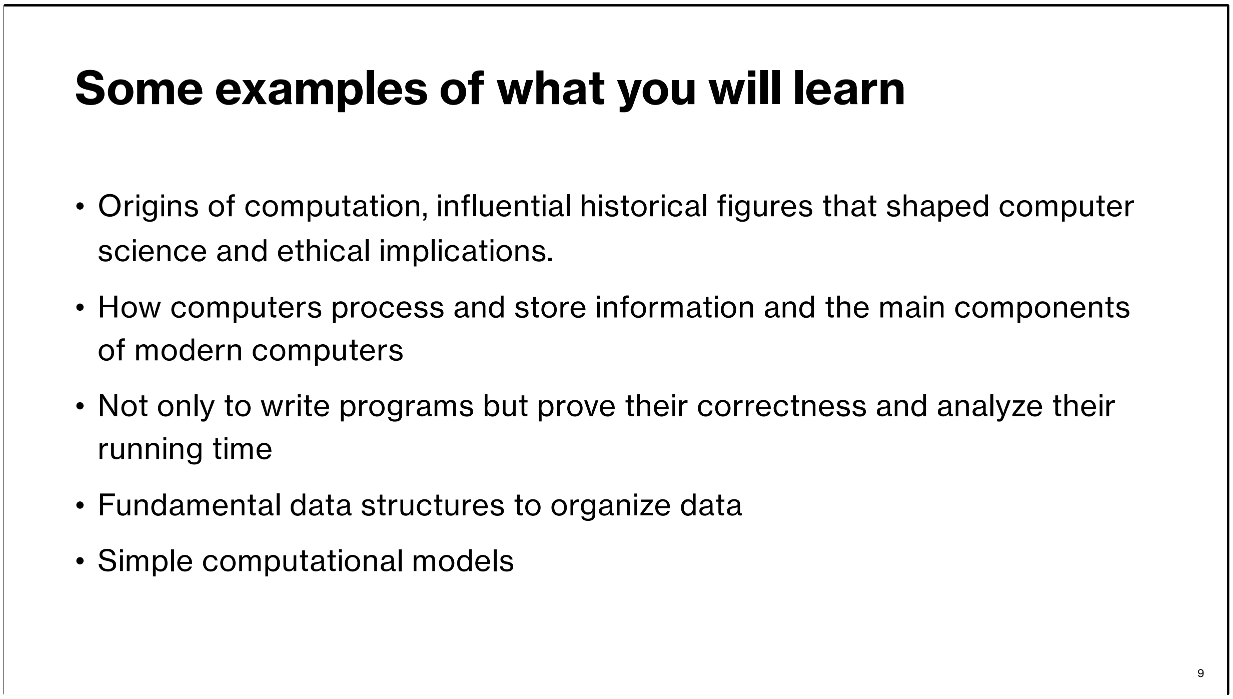
Now turn to your neighbor, make sure you know each other's name and discuss the following two prompts.
When do you think the first computer was invented and which people come to mind when thinking of influential figures, both historic and modern, when thinking of technology?
Great, these are going to be two topics we will tackle today.



What is CS51?

- An introduction to computer science through a survey of fundamental topics and a glimpse into our curriculum.
  - Week 1: History and Ethics of Computer Science
  - Weeks 2-5: Computer Systems (CSCI105)
  - Weeks 6-9: Mathematical Foundations of Computer Science (CSCI054)
  - Weeks 10-13: Data Structures and Algorithms (CSCI062 and CSCI140)
  - Weeks 13-15: Theory of Computation & Programming Languages (CS101)
  - Weeks 15-16: Machine Learning (electives)

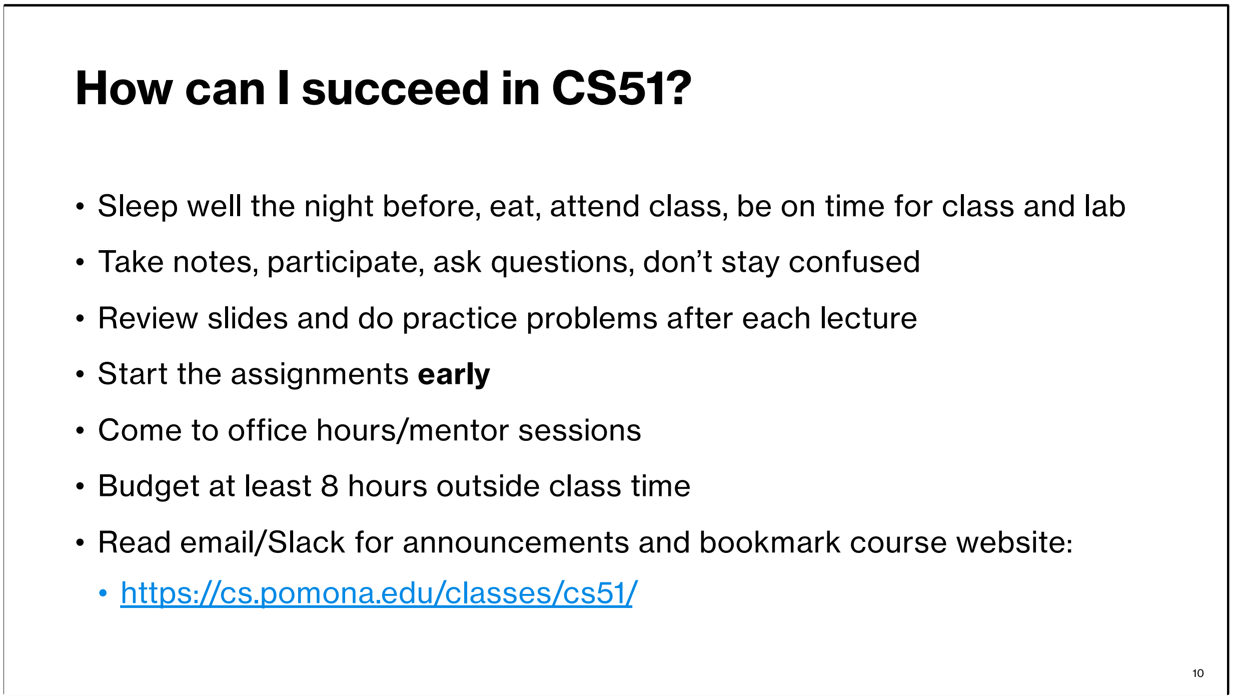
CS51 has been designed as an introduction to the field of computer science at large, where programming is just one component and not the whole focus. Together, we will survey some of the biggest areas of computer science. This will also help you get a glimpse of what to expect in our intro sequence and upper division courses, if you choose to continue taking classes in our department. Today, we will start with an introduction to the history of cs. Here is what to expect for the rest of the semester.



Some examples of what you will learn

- Origins of computation, influential historical figures that shaped computer science and ethical implications.
- How computers process and store information and the main components of modern computers
- Not only to write programs but prove their correctness and analyze their running time
- Fundamental data structures to organize data
- Simple computational models

Here are some examples of what we will learn this semester. You will exposed to the origins of computation and learn about important historical figures that have shaped the field of computer science and what technology looks like today
You will understand that everything we do as computer scientists has the potential to impact society and become more comfortable with thinking of the ethics of technology
You will become familiar with how computers process and store information, what the main components of your computer are and how they work together
You will see how computer scientists not only write programs but can prove their correctness
You will learn fundamental data structures to organize data
You will be exposed to the idea that the algorithms we design should not only be correct but efficient and that depending on the situation we might choose different algorithms
You will see that the idea of computation can be reduced to seemingly simple machines that have the same capabilities with today’s computers

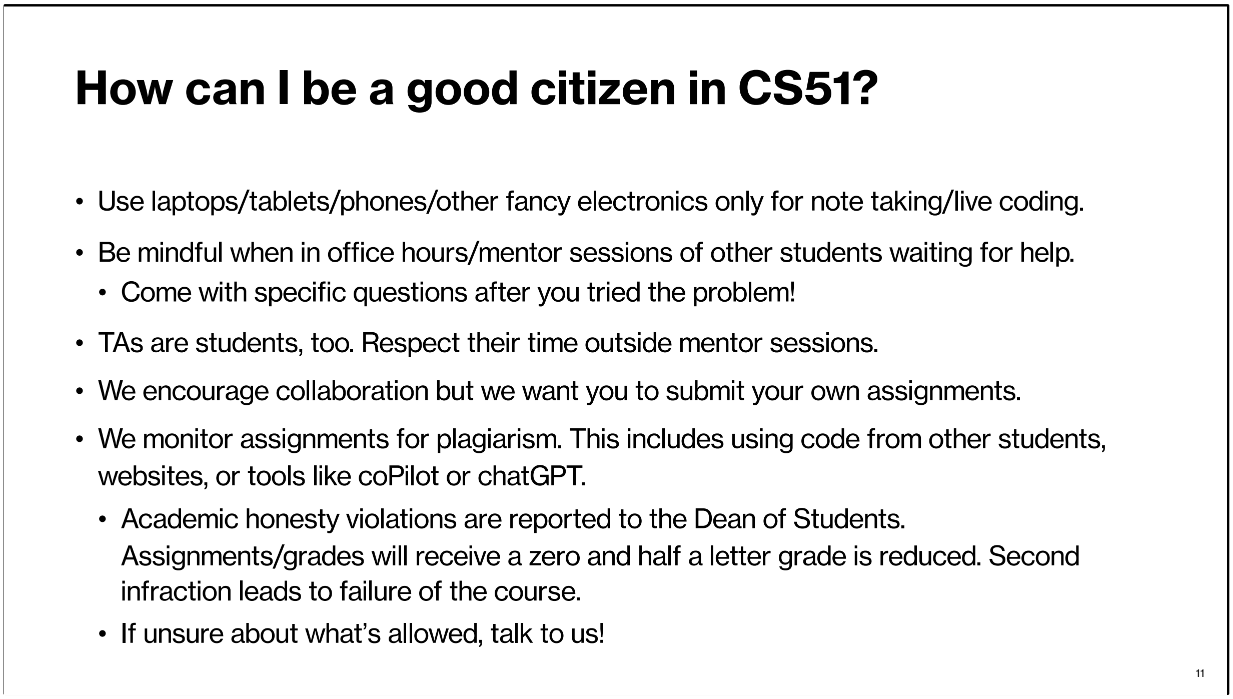


How can I succeed in CS51?

- Sleep well the night before, eat, attend class, be on time for class and lab
- Take notes, participate, ask questions, don’t stay confused
- Review slides and do practice problems after each lecture
- Start the assignments **early**
- Come to office hours/mentor sessions
- Budget at least 8 hours outside class time
- Read email/Slack for announcements and bookmark course website:
  - https://cs.pomona.edu/classes/cs51/

10

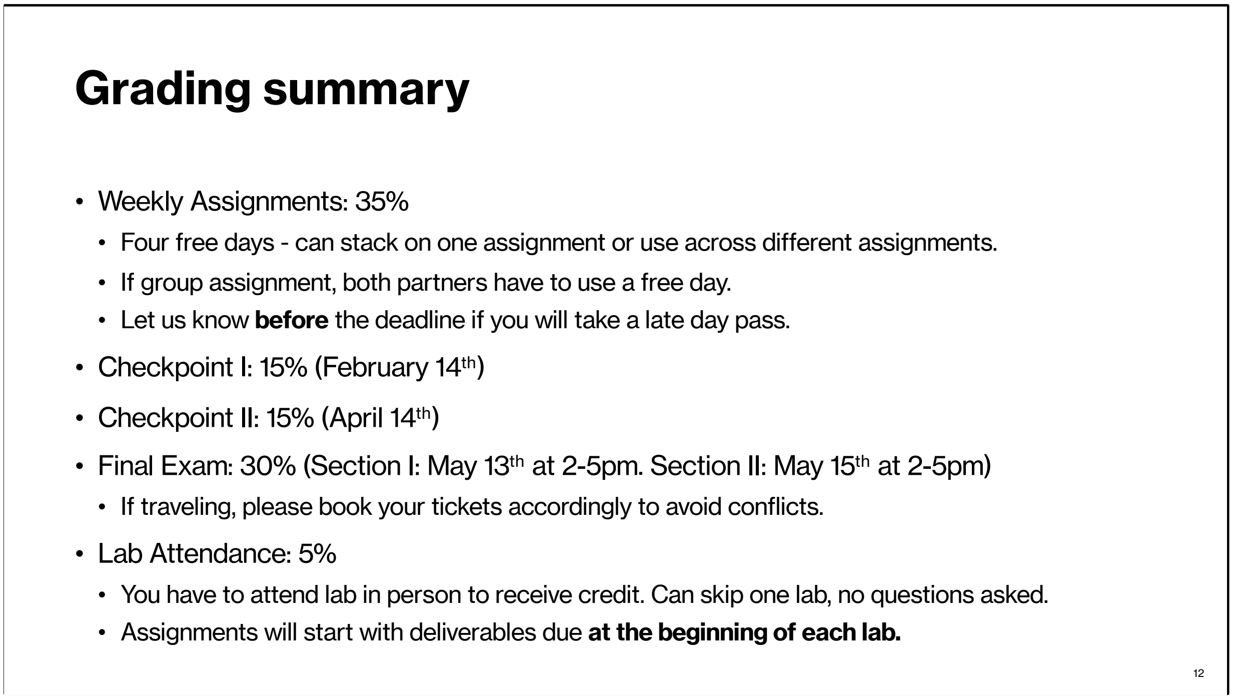
How can you do well in this course (and in College in general)? I always tell students to not underestimate the importance of a good night’s sleep and eating well. We expect you to be here and in lab on time to avoid disruptions and make the most of the time we have together. The slides are available online along with notes, but feel free to add your own notes on them. We want you to participate, ask questions, and not stay confused. It’s always a good idea to review what we covered in class right after since it will be fresh in your mind. The class will have 10 weekly assignments. I can’t stretch enough how important it is to start early. The entire course staff is here with office hours and mentor sessions to help you. But we are asking you to try the problem first before you come to us. This is a full-credit course; please budget at least 8 hours outside class time. And don’t forget to keep an eye on email and Slack for announcements as well as regularly check the website for the slides and assignments.



How can I be a good citizen in CS51?

- Use laptops/tablets/phones/other fancy electronics only for note taking/live coding.
- Be mindful when in office hours/mentor sessions of other students waiting for help.
  - Come with specific questions after you tried the problem!
- TAs are students, too. Respect their time outside mentor sessions.
- We encourage collaboration but we want you to submit your own assignments.
- We monitor assignments for plagiarism. This includes using code from other students, websites, or tools like coPilot or chatGPT.
  - Academic honesty violations are reported to the Dean of Students. Assignments/grades will receive a zero and half a letter grade is reduced. Second infraction leads to failure of the course.
  - If unsure about what's allowed, talk to us!

If you choose to use electronics, please do so only for note taking and live coding. When coming to office hours and mentor sessions, please come with specific questions and respect that TAs are students, too, that are working a job with specific hours. Although I am happy to see students talking about assignments, I expect that the work you submit is your own. This means that you cannot use tools like copilot or chatgpt. This is an intentional policy for an introductory course in cs. Such tools can make you more efficient but they both take away from the necessary part of struggling to fully own knowledge, and circumvent crucial skills you will need both as a potential computer scientists and college student in general. Please take this seriously. Academic honesty violations are a serious matter and they get reported to the Dean of Students with consequences both for this class and future employment. If you are unsure, please come and talk to us.



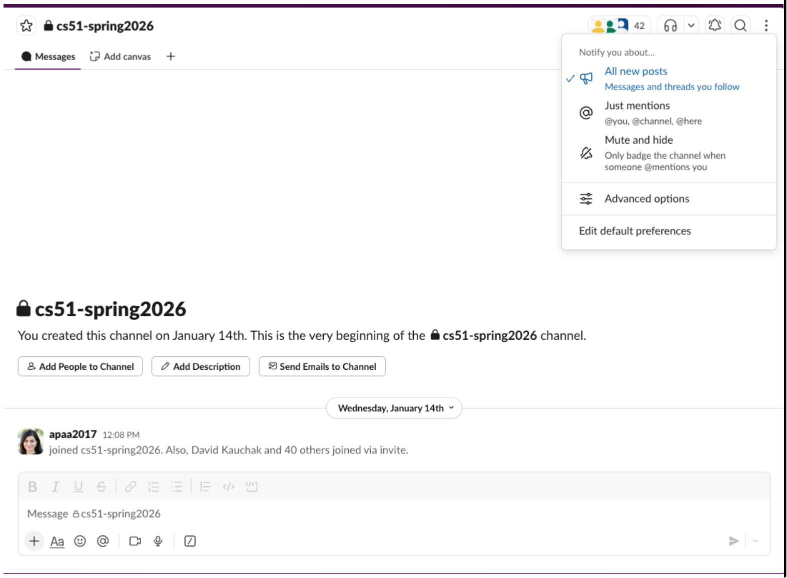
Grading summary

- Weekly Assignments: 35%
  - Four free days - can stack on one assignment or use across different assignments.
  - If group assignment, both partners have to use a free day.
  - Let us know **before** the deadline if you will take a late day pass.
- Checkpoint I: 15% (February 14th)
- Checkpoint II: 15% (April 14th)
- Final Exam: 30% (Section I: May 13th at 2-5pm. Section II: May 15th at 2-5pm)
  - If traveling, please book your tickets accordingly to avoid conflicts.
- Lab Attendance: 5%
  - You have to attend lab in person to receive credit. Can skip one lab, no questions asked.
  - Assignments will start with deliverables due **at the beginning of each lab.**

Here is a breakdown of what to expect for grading. There are ten weekly assignments: 35%. You get four free days which you can use on one assignment or across different assignments. Let me know **before** the deadline if you will take a late day pass. If it is a group assignment, both partners have to use a free day
Two checkpoints, each 15% and a Final Exam: 30%
Labs Attendance: 5%. You have to attend lab in person to receive credit. You can skip *one* lab, no questions asked

Slack channels

- If registered, you have been invited to cs51-spring2026 channel in http://slack.pomona.edu
- Department-wide Slack workspace https://tinyurl.com/PomonaCSSlack



cs51-spring2026

You created this channel on January 14th. This is the very beginning of the cs51-spring2026 channel.

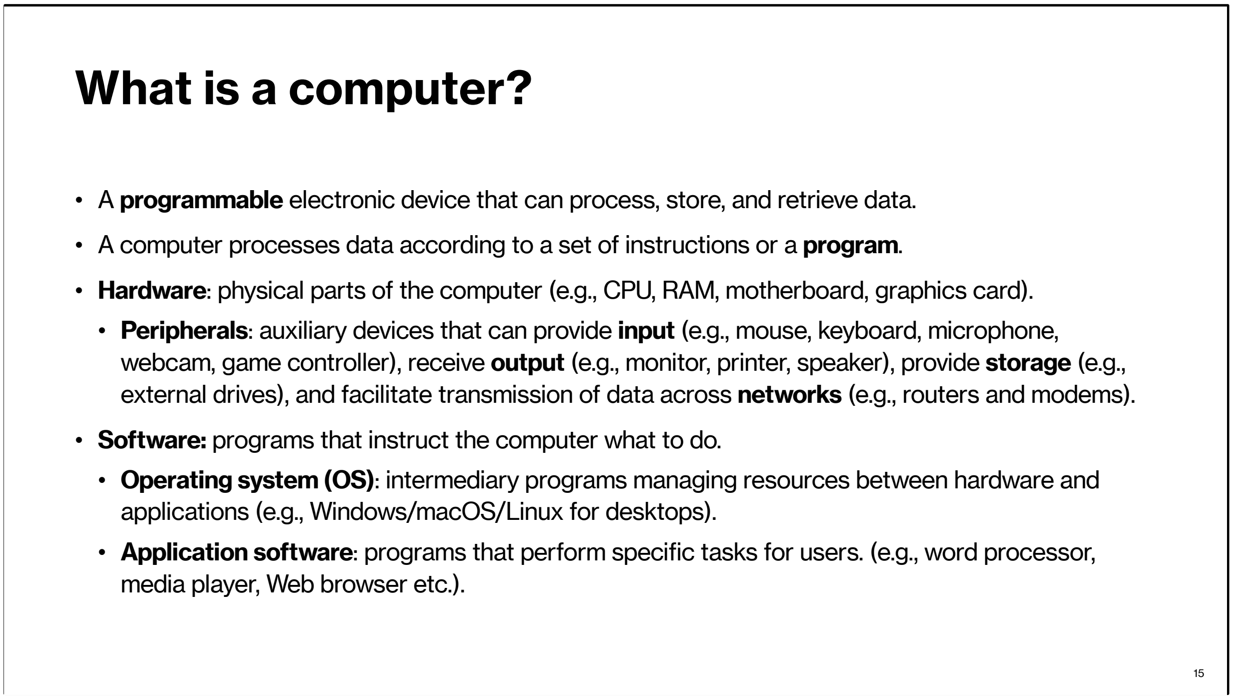
apaa2017 12:08 PM
joined cs51-spring2026. Also, David Kauchak and 40 others joined via invite.

For the class, we will use the cs51-spring2026 slack channel to answer questions off-class and to send announcements, e.g. when assignments have been graded, reminders for mentor sessions, impromptu changes to office hours or mentor sessions. If you are registered in CS51, I have already invited you to this channel. Please let me know if you are facing any issues with it after class. You can post questions with your name known to everyone or anonymously. Please make sure that the notifications are on.

There's also a departmental slack workspace with multiple channels that can help you socialize with other people in the department.

Computer Science History

Unless there are any more questions on course logistics, we are ready to move to actual content material where we will start talking about the History of Computer Science as a field.



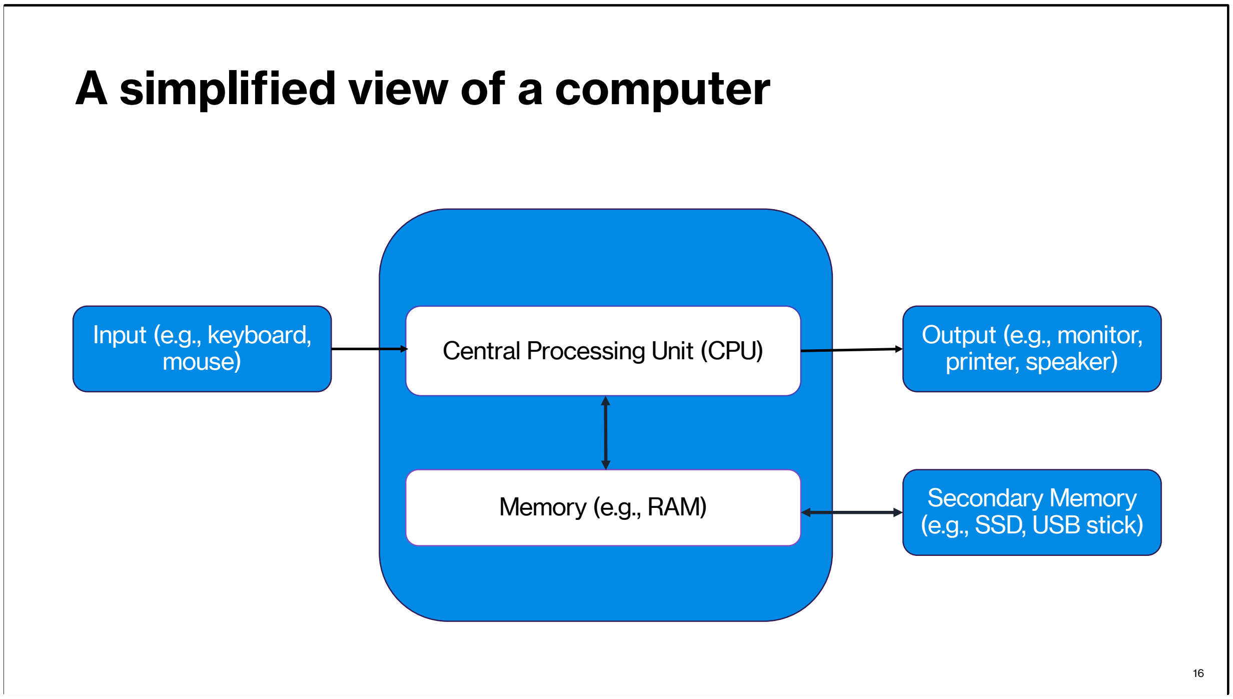
What is a computer?

- A **programmable** electronic device that can process, store, and retrieve data.
- A computer processes data according to a set of instructions or a **program**.
- **Hardware**: physical parts of the computer (e.g., CPU, RAM, motherboard, graphics card).
  - **Peripherals**: auxiliary devices that can provide **input** (e.g., mouse, keyboard, microphone, webcam, game controller), receive **output** (e.g., monitor, printer, speaker), provide **storage** (e.g., external drives), and facilitate transmission of data across **networks** (e.g., routers and modems).
- **Software:** programs that instruct the computer what to do.
  - **Operating system (OS)**: intermediary programs managing resources between hardware and applications (e.g., Windows/macOS/Linux for desktops).
  - **Application software**: programs that perform specific tasks for users. (e.g., word processor, media player, Web browser etc.).

Computers are an integral part of modern society and new technology has transformed the world into an interconnected and highly interdependent global community. One example of how much computers have turned the world into a global village is that of communication: messages, phones, calls, video conferencing, email, social media, are all examples of how computers and networks bring us together. We will see a lot of the negative aspects of this proliferation of computers during the next class meeting. But it is indisputable that computers are behind most aspects of commerce, communication, transportation, health, education, entertainment and management of public and private infrastructure.

Today, I want us to establish some facts about the history of computers starting with what is a computer?
When we talk about computers, we understand them as electronic devices that we can program and which can process, store, and retrieve data. A computer processes data according to a set of instructions or a program.

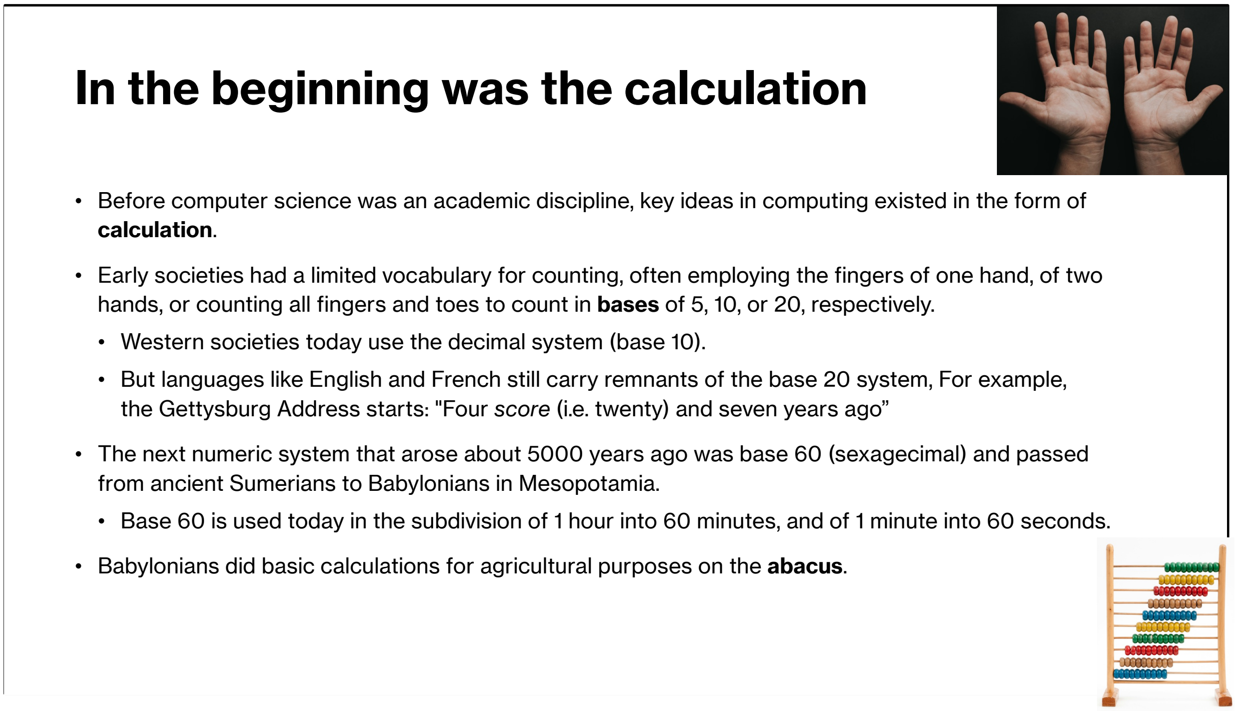
Computers consist of two parts: hardware (including peripheral devices) and software. Hardware is the physical parts of the computer. If you ever have seen the “guts” of a desktop computer tower that would be its hardware; think of its CPU, RAM, motherboard, graphics card, etc. Peripherals are auxiliary devices that provide input, output, storage, and networking capabilities, like the mouse, keyboard, webcam, monitor, etc. Software is all the programs that instruct the computer what to do. Largely speaking that is either the operating system, which sits between the hardware and applications, and application software which performs specific tasks for the user. What operating system does your computer run on? It’s likely Windows, macOS, or some variation of Linux. What are your favorite examples of application software that you use? Mine has to be PowerPoint these days.



A simplified view of a computer

Input (e.g., keyboard, mouse)
Central Processing Unit (CPU)
Output (e.g., monitor, printer, speaker)
Memory (e.g., RAM)
Secondary Memory (e.g., SSD, USB stick)

A simplified view of a modern computer would show that it comprises mostly of two components. A central processing unit, i.e. the CPU, and memory (also known as primary or main memory). You can think of the CPU as the brains of your computer that communicate with the memory and auxiliary devices. The memory stores current data, programs, and instructions that can be accessed directly by the CPU. Any data that should not be changed is stored in a special type of main memory called ROM (read-only memory). For example, the programs that your operating system needs when it boots are stored in ROM. In contrast, RAM (random-access memory) is like our computer's short-term memory for data and programs that need to be read and written but is volatile, that is its information is wiped if power is removed. Secondary memory stores programs and data permanently, without having to worry about what happens when we power off our computer. Secondary memory comes mostly in the form of solid-state drives (SSDs) these days. They are smaller, faster, and less noisy than the older hard-disk drives (HDDs). Other examples of secondary memory are usb drives, memory cards, CDs/DVDs. All this exchange of information can be supplemented by input that the user provides through their keyboard, mouse, etc, and it can be directed to output devices such as the monitor, printers, speakers, etc.



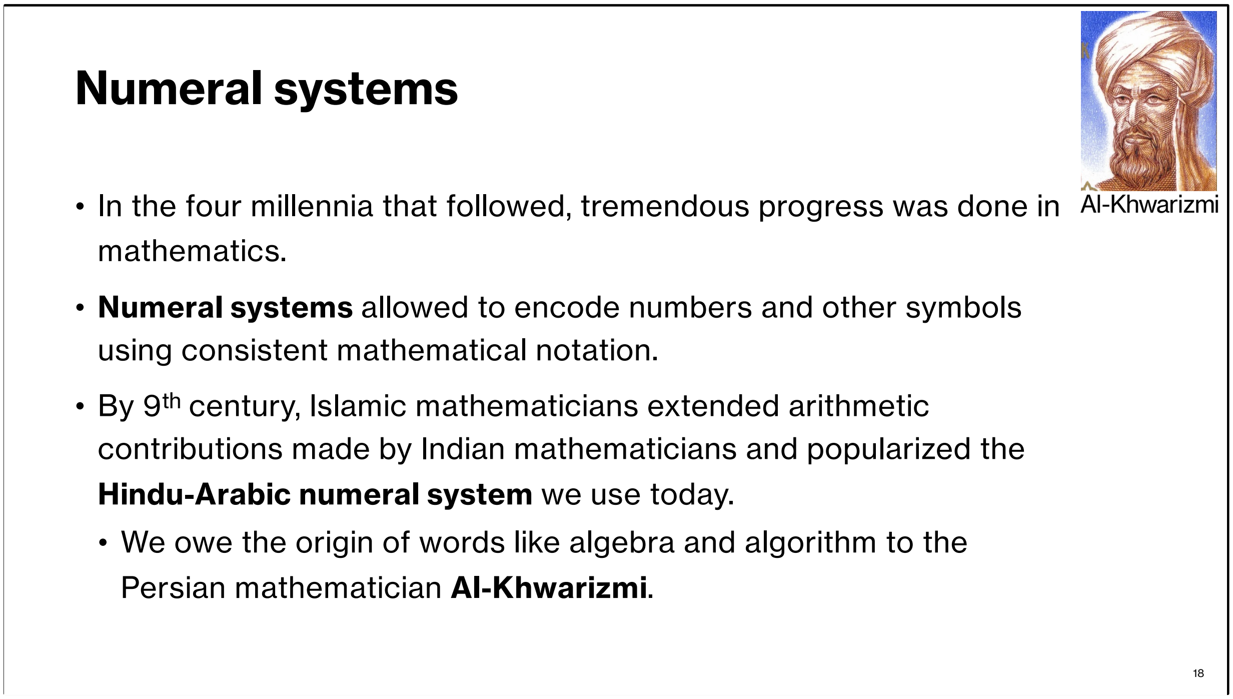
In the beginning was the calculation

- Before computer science was an academic discipline, key ideas in computing existed in the form of **calculation**.
- Early societies had a limited vocabulary for counting, often employing the fingers of one hand, of two hands, or counting all fingers and toes to count in **bases** of 5, 10, or 20, respectively.
  - Western societies today use the decimal system (base 10).
  - But languages like English and French still carry remnants of the base 20 system, For example, the Gettysburg Address starts: "Four *score* (i.e. twenty) and seven years ago”
- The next numeric system that arose about 5000 years ago was base 60 (sexagecimal) and passed from ancient Sumerians to Babylonians in Mesopotamia.
  - Base 60 is used today in the subdivision of 1 hour into 60 minutes, and of 1 minute into 60 seconds.
- Babylonians did basic calculations for agricultural purposes on the **abacus**.

Long before the advent of the modern computer and before computer science was an academic field, especially one worth studying, the ideas in computing were subtly woven through human civilization in the form of **calculation**. Early societies had a limited vocabulary for counting, often employing the fingers of one hand, of two hands, or counting all fingers and toes to count in **bases** of 5, 10, or 20, respectively. Western societies today use the decimal system (base 10): we count as 0, 1, 2, 3, ...9, and so on. But languages like English and French still carry remnants of the base 20 system, e.g., The Gettysburg Address starts: "Four *score* (i.e. twenty) and seven years ago” or quatre *vingt* for eighty (4x20), respectively.

The next numeric system that arose about 5000 years ago was base 60 (sexagecimal) and passed from ancient

Sumerians to Babylonians in Mesopotamia (ancient Iraq). Base 60 is still used today in the subdivision of one hour into 60 minutes, and of one minute into 60 seconds. The Babylonians did basic calculations for agricultural purposes, such as adding and subtracting crops and livestock, on the **abacus**, the earliest recognized device for computing. You might have encountered it in your early years of education, when sliding colorful beams on rods to learn how to add numbers.



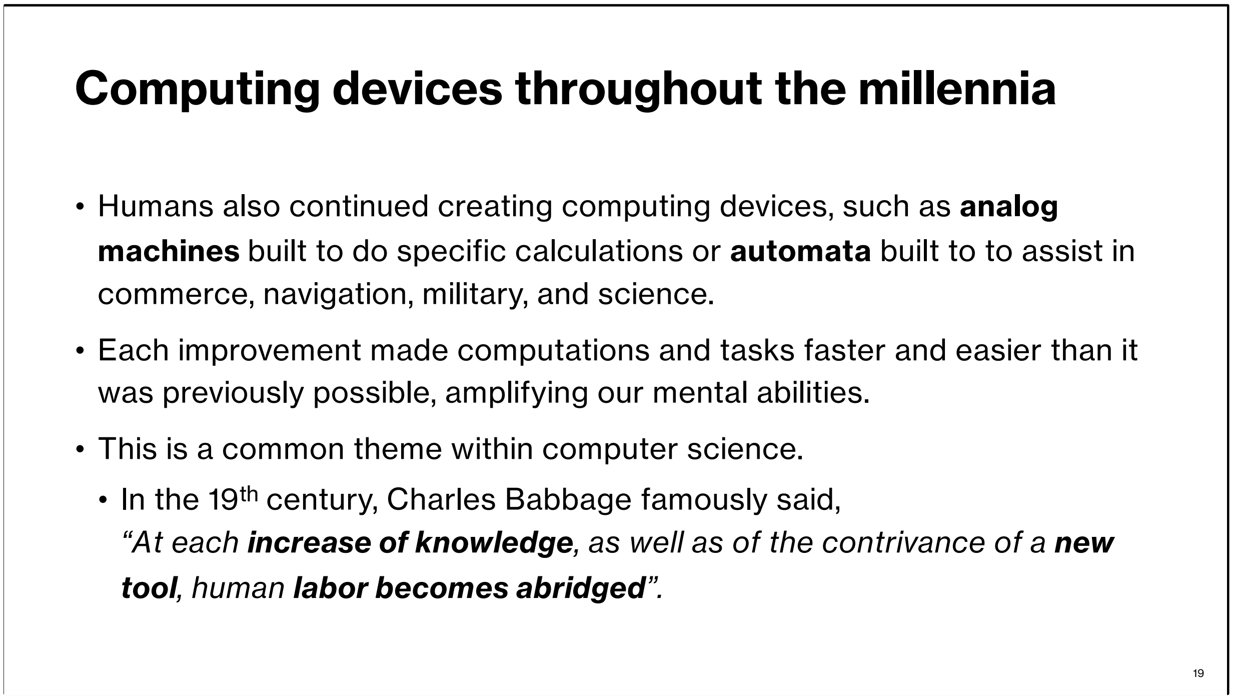
Numeral systems

Al-Khwarizmi

- In the four millennia that followed, tremendous progress was done in mathematics.
- **Numeral systems** allowed to encode numbers and other symbols using consistent mathematical notation.
- By 9th century, Islamic mathematicians extended arithmetic contributions made by Indian mathematicians and popularized the **Hindu-Arabic numeral system** we use today.
  - We owe the origin of words like algebra and algorithm to the Persian mathematician **Al-Khwarizmi**.

18

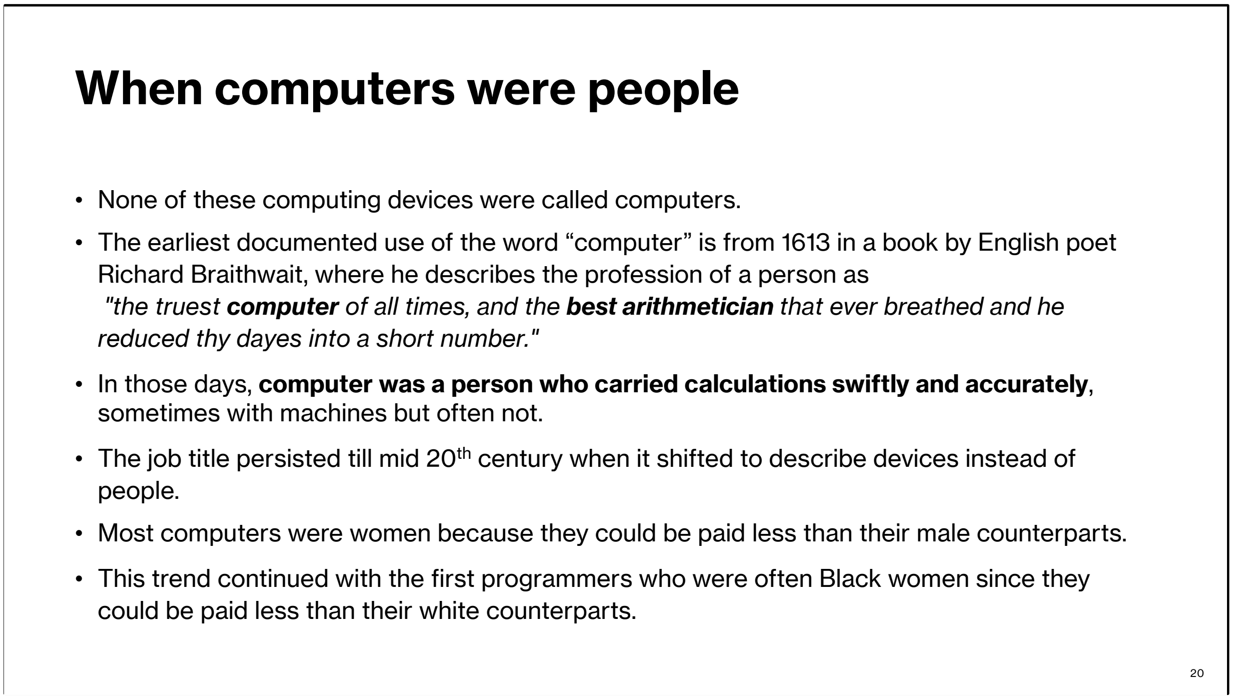
In the four millennia that followed, tremendous progress was done in mathematics, including geometry, algebra, and systems of proofs. The creation of numeral systems allowed humans to encode numbers and other symbols using consistent mathematical notation. By 9th century, Islamic mathematicians extended arithmetic contributions made by Indian mathematicians and popularized the Hindu-Arabic numeral system we use today. We owe the origin of words like algebra and algorithm to the Persian mathematician Al-Khwarizmi pictured above.



Computing devices throughout the millennia

- Humans also continued creating computing devices, such as **analog machines** built to do specific calculations or **automata** built to to assist in commerce, navigation, military, and science.
- Each improvement made computations and tasks faster and easier than it was previously possible, amplifying our mental abilities.
- This is a common theme within computer science.
  - In the 19th century, Charles Babbage famously said, *"At each **increase of knowledge**, as well as of the contrivance of a **new tool**, human **labor becomes abridged**".*

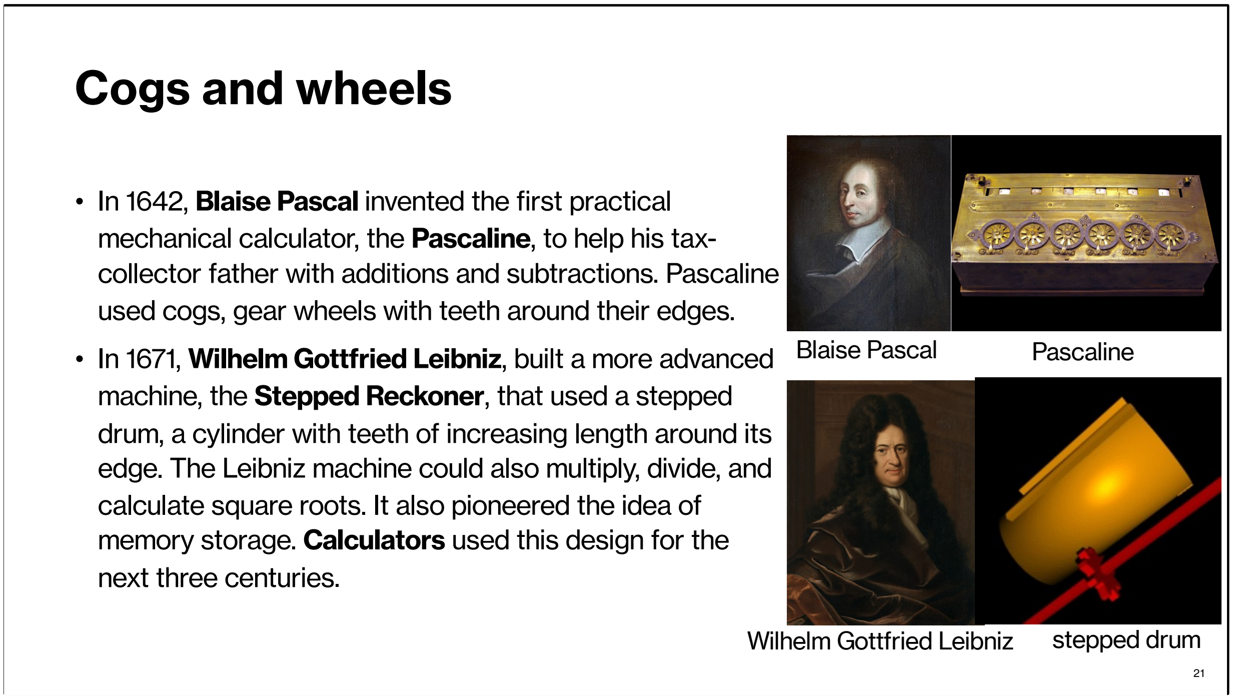
At the same time, humans continued making contributions in mathematics and devised computing devices, such as analog machines built to do specific calculations or automata built to accomplish certain tasks to assist in commerce, navigation, military, and science. Each improvement made computations and tasks faster and easier than it was previously possible, augmenting our mental abilities. Note that this is a common theme in computer science. In the 19th century, early computer pioneer Charles Babbage (we'll see more of him soon) famously said, "At each increase of knowledge, as well as of the contrivance of a new tool, human labor becomes abridged".



When computers were people

- None of these computing devices were called computers.
- The earliest documented use of the word “computer” is from 1613 in a book by English poet Richard Braithwait, where he describes the profession of a person as *"the truest **computer** of all times, and the **best arithmetician** that ever breathed and he reduced thy dayes into a short number."*
- In those days, **computer was a person who carried calculations swiftly and accurately**, sometimes with machines but often not.
- The job title persisted till mid 20th century when it shifted to describe devices instead of people.
- Most computers were women because they could be paid less than their male counterparts.
- This trend continued with the first programmers who were often Black women since they could be paid less than their white counterparts.

Nevertheless, none of these computing devices were called computers. The earliest documented use of the word “computer” is from 1613 in the book *”The Yong Mans Gleanings”* by English poet Richard Braithwait, where he describes the profession of a person as *"the truest computer of all times, and the best arithmetician that ever breathed and he reduced thy dayes into a short number"*. In those days, computer was a person who carried calculations swiftly and accurately, sometimes with machines but often not. The job title persisted till mid 20th century where it had now shifted to describe devices instead of people. Interestingly, most computers were women because they could be paid less than their male counterparts. This trend continued with the first programmers who were often Black women since they could be paid less than their white counterparts.



Cogs and wheels

- In 1642, **Blaise Pascal** invented the first practical mechanical calculator, the **Pascaline**, to help his tax-collector father with additions and subtractions. Pascaline used cogs, gear wheels with teeth around their edges.
- In 1671, **Wilhelm Gottfried Leibniz**, built a more advanced machine, the **Stepped Reckoner**, that used a stepped drum, a cylinder with teeth of increasing length around its edge. The Leibniz machine could also multiply, divide, and calculate square roots. It also pioneered the idea of memory storage. **Calculators** used this design for the next three centuries.

Blaise Pascal

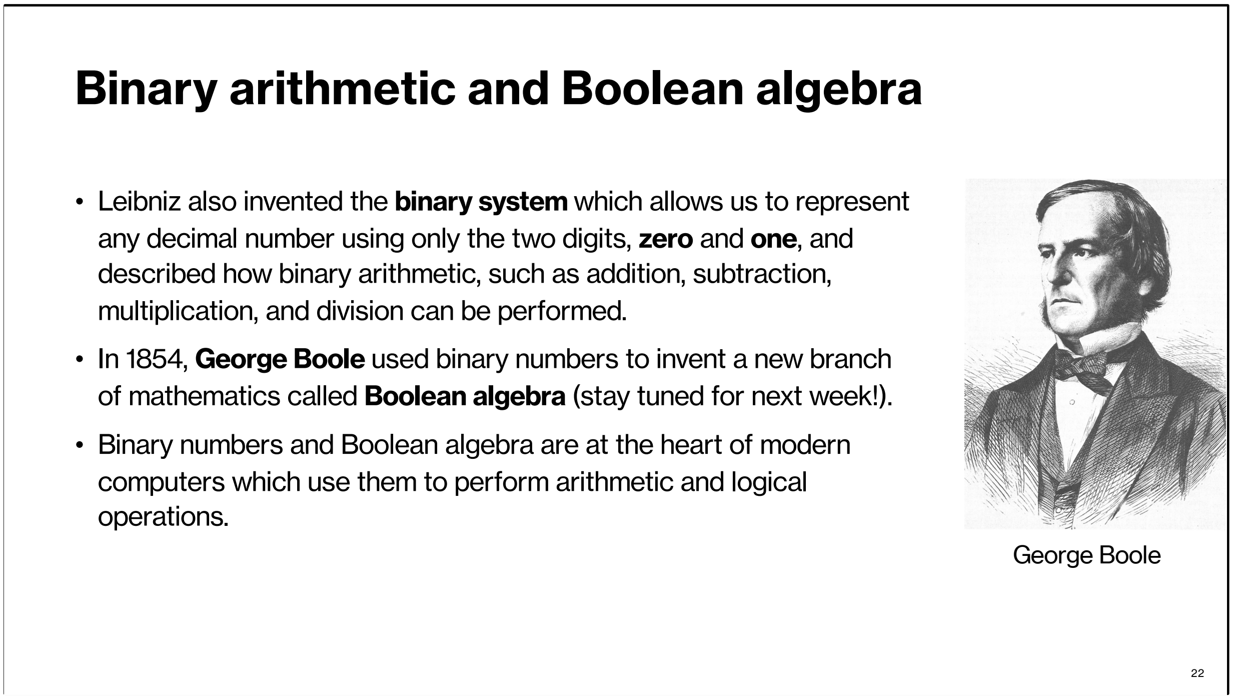
Pascaline

Wilhelm Gottfried Leibniz

stepped drum

21

Going back to computing devices, here are two notable examples. in 1642, aged only 18, French scientist and philosopher Blaise Pascal (1623–1666) invented the first practical mechanical calculator, the Pascaline, to help his tax-collector father do his sums. The machine had a series of interlocking cogs (gear wheels with teeth around their outer edges) that could add and subtract decimal numbers. Several decades later, in 1671, German mathematician and philosopher Gottfried Wilhelm Leibniz (1646–1716) came up with a similar but more advanced machine, the Stepped Reckoner. Instead of using cogs, it had a "stepped drum" (a cylinder with teeth of increasing length around its edge). The Leibniz machine could do much more than Pascal's: as well as adding and subtracting, it could multiply, divide, and work out square roots. Another pioneering feature was the first memory store or "register." The design was so successful that calculators used it for the next three centuries.

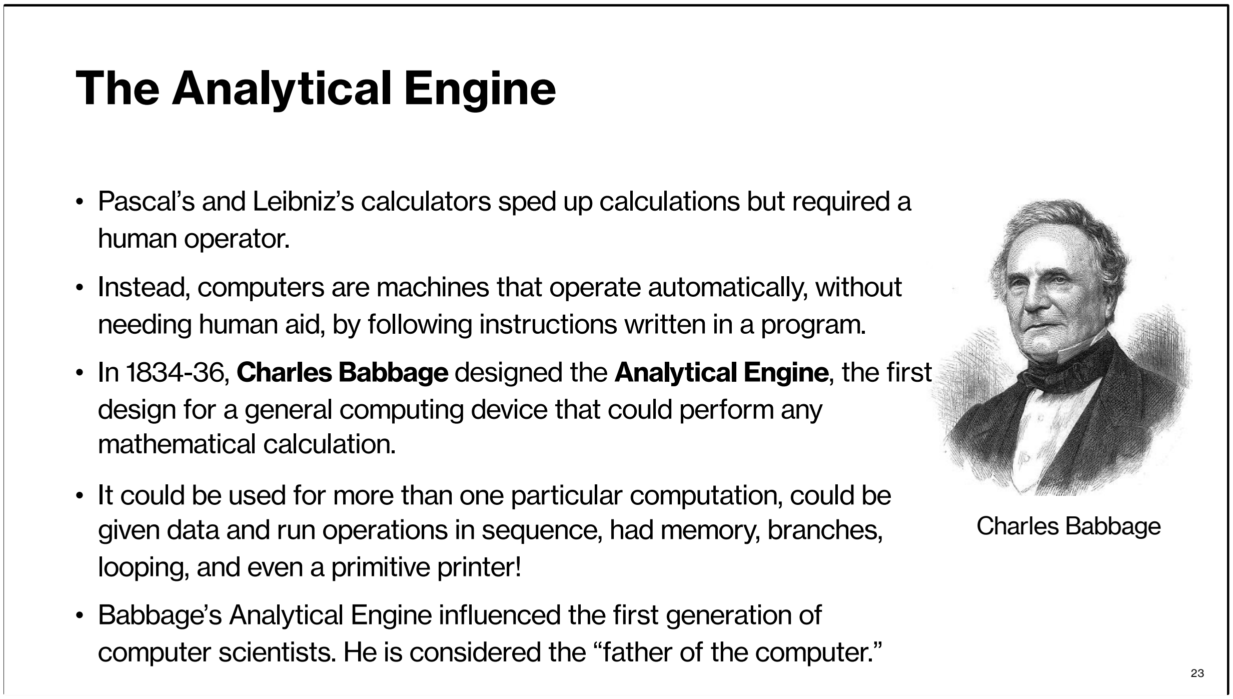


Binary arithmetic and Boolean algebra

- Leibniz also invented the **binary system** which allows us to represent any decimal number using only the two digits, **zero** and **one**, and described how binary arithmetic, such as addition, subtraction, multiplication, and division can be performed.
- In 1854, **George Boole** used binary numbers to invent a new branch of mathematics called **Boolean algebra** (stay tuned for next week!).
- Binary numbers and Boolean algebra are at the heart of modern computers which use them to perform arithmetic and logical operations.

George Boole

Apart from developing one of the world's earliest mechanical calculators, Leibniz is remembered for another important contribution to computing: he was the man who invented binary code, a way of representing any decimal number using only the two digits zero and one. He also described how to use binary numbers to do basic arithmetic such as addition and subtraction. Although Leibniz made no use of binary in his own calculator, it set others thinking. In 1854, a little over a century after Leibniz had died, Englishman George Boole (1815–1864) used the idea to invent a new branch of mathematics called Boolean algebra. In modern computers, binary code and Boolean algebra allow computers to make simple decisions by comparing long sequences of zeros and ones. But, in the 19th century, these ideas were still far ahead of their time. It would take another 50–100 years for mathematicians and computer scientists to figure out how to use them.

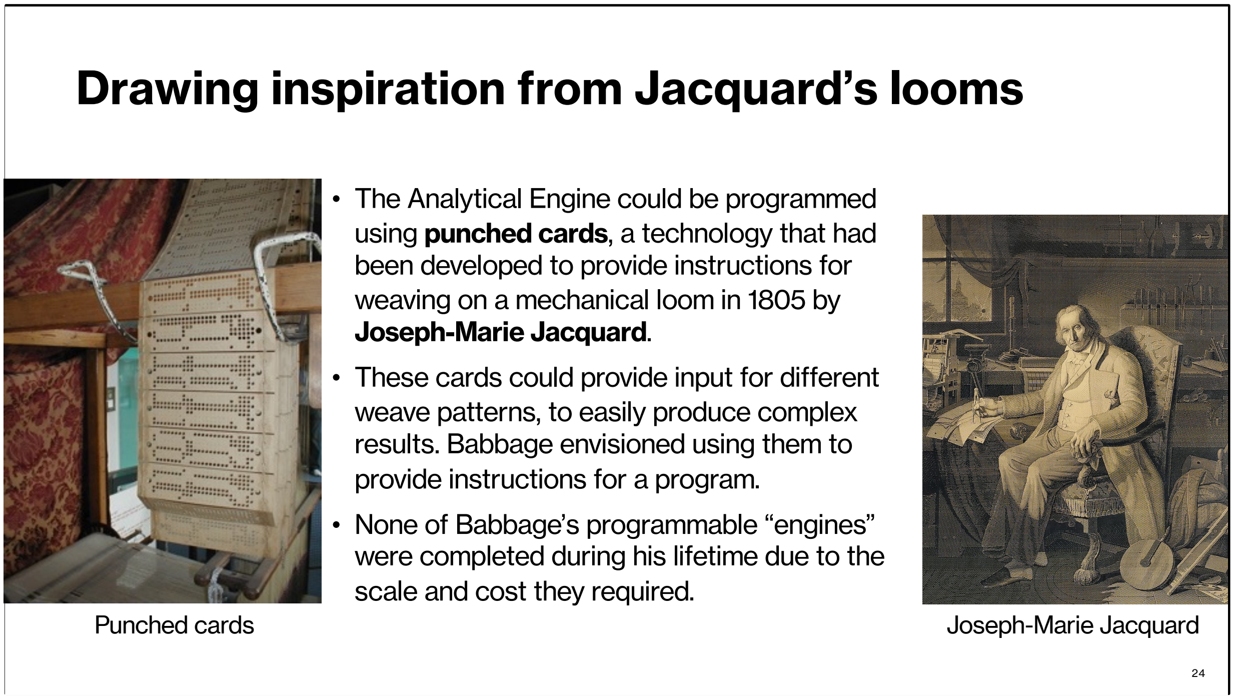


The Analytical Engine

- Pascal’s and Leibniz’s calculators sped up calculations but required a human operator.
- Instead, computers are machines that operate automatically, without needing human aid, by following instructions written in a program.
- In 1834-36, **Charles Babbage** designed the **Analytical Engine**, the first design for a general computing device that could perform any mathematical calculation.
- It could be used for more than one particular computation, could be given data and run operations in sequence, had memory, branches, looping, and even a primitive printer!
- Babbage’s Analytical Engine influenced the first generation of computer scientists. He is considered the “father of the computer.”

Charles Babbage

Pascal’s and Leibniz’s calculators sped up calculations but required a human operator. Instead, computers are machines that operate automatically, without needing human aid, by following instructions written in a program. In 1834-36, Charles Babbage designed the Analytical Engine, the first design for a general computing device that could perform any mathematical calculation. Babbage's design could be used for more than one particular computation, could be given data and run operations in sequence, had memory, branches, looping, and even a primitive printer! Babbage’s design of the Analytical Engine influenced the first generation of computer scientists. He is considered the “father of the computer.”



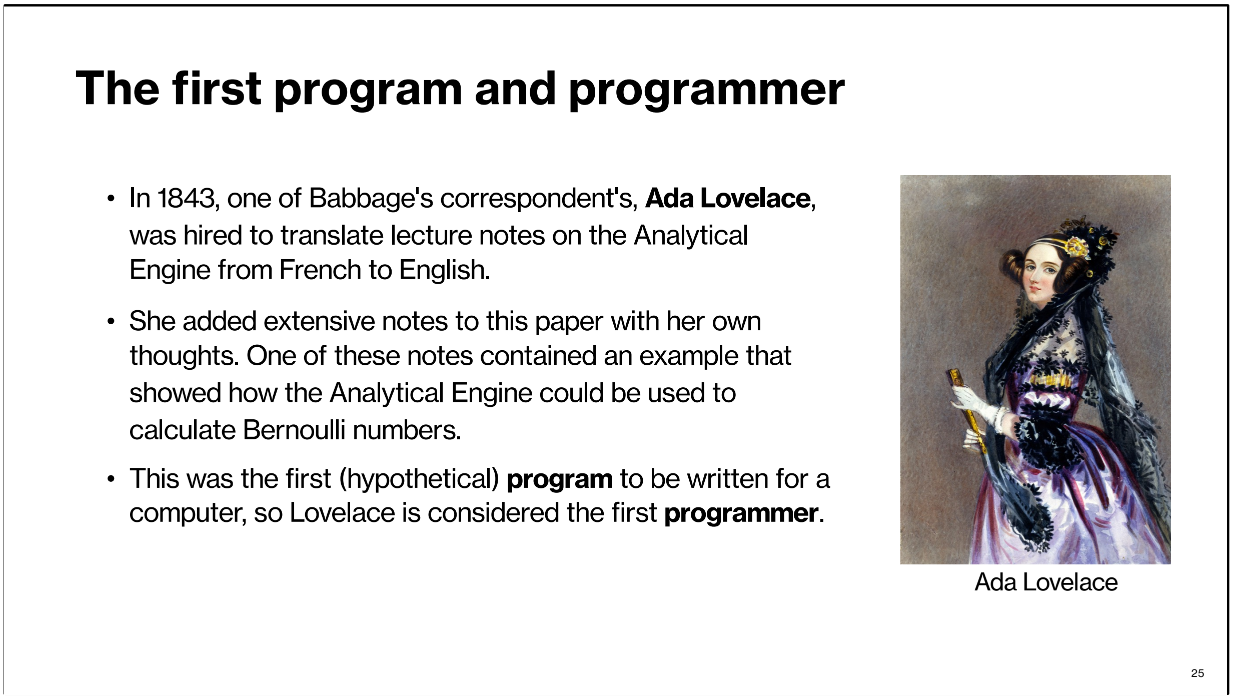
Drawing inspiration from Jacquard’s looms

- The Analytical Engine could be programmed using **punched cards**, a technology that had been developed to provide instructions for weaving on a mechanical loom in 1805 by **Joseph-Marie Jacquard**.
- These cards could provide input for different weave patterns, to easily produce complex results. Babbage envisioned using them to provide instructions for a program.
- None of Babbage’s programmable “engines” were completed during his lifetime due to the scale and cost they required.

Punched cards

Joseph-Marie Jacquard

The Analytical Engine could be programmed using 'punched cards’, paper-based cards with a grid of locations that can be punched out to represent data and which been developed to provide instructions for weaving on a mechanical loom in 1805 by French weaver and merchant Joseph-Marie Jacquard (1752-1834). These cards could provide input for different weave patterns, to easily produce complex results. Babbage envisioned using them to provide instructions for a program. None of Babbage’s programmable “engines” were completed during his lifetime due to the scale and cost they required but eventually in the 1990s am earlier version, the difference engine, was built.



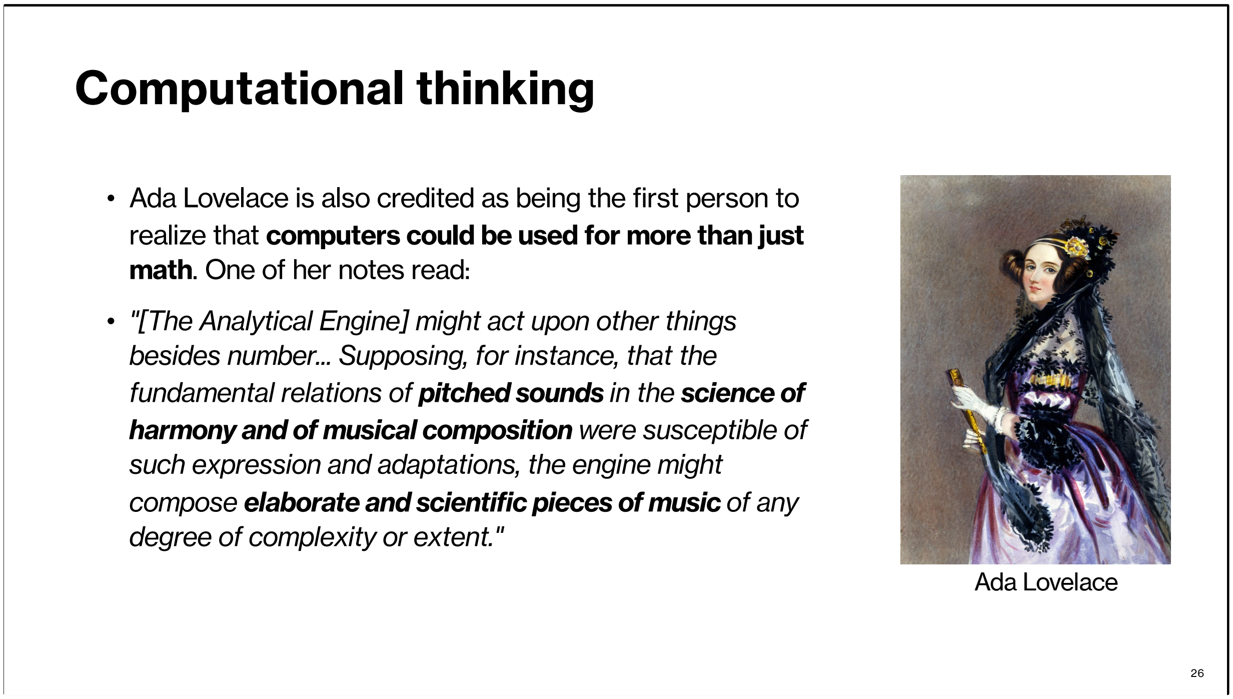
The first program and programmer

- In 1843, one of Babbage's correspondent's, **Ada Lovelace**, was hired to translate lecture notes on the Analytical Engine from French to English.
- She added extensive notes to this paper with her own thoughts. One of these notes contained an example that showed how the Analytical Engine could be used to calculate Bernoulli numbers.
- This was the first (hypothetical) **program** to be written for a computer, so Lovelace is considered the first **programmer**.

Ada Lovelace

25

Babbage was more fortunate in receiving help from Augusta Ada Byron (1815–1852), Countess of Lovelace, daughter of the poet Lord Byron. An enthusiastic mathematician, Ada Lovelace added extensive notes to her translation of a French manual of the mechanics of the Analytical Engine. One of these notes contained an example that showed how the Analytical Engine could be used to calculate Bernoulli numbers. This is considered the first program to be written for a computer and Lovelace is referred to as the world's first computer programmer.

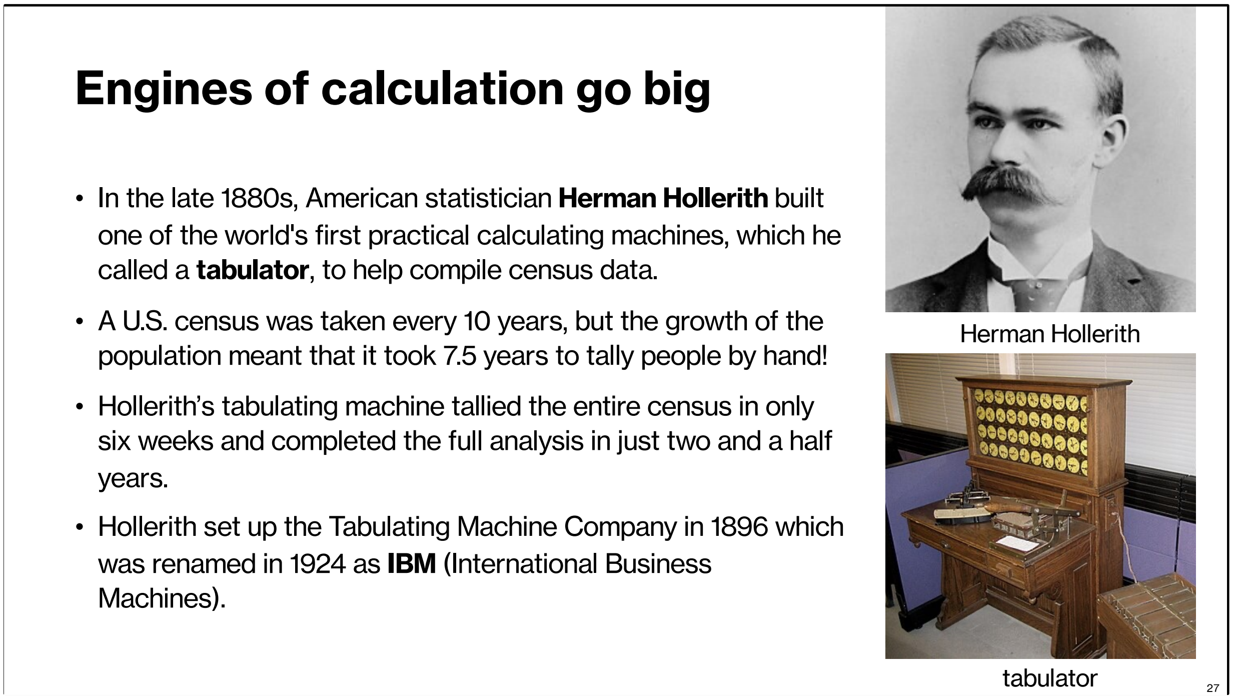


Computational thinking

- Ada Lovelace is also credited as being the first person to realize that **computers could be used for more than just math**. One of her notes read:
- *"[The Analytical Engine] might act upon other things besides number... Supposing, for instance, that the fundamental relations of* ***pitched sounds*** *in the* ***science of harmony and of musical composition*** *were susceptible of such expression and adaptations, the engine might compose* ***elaborate and scientific pieces of music*** *of any degree of complexity or extent."*

Ada Lovelace

Ada Lovelace is also credited as being the first person to realize that computers could be used for more than just doing mathematic calculations. One of her notes read: "[The Analytical Engine] might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine... Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent."



Engines of calculation go big

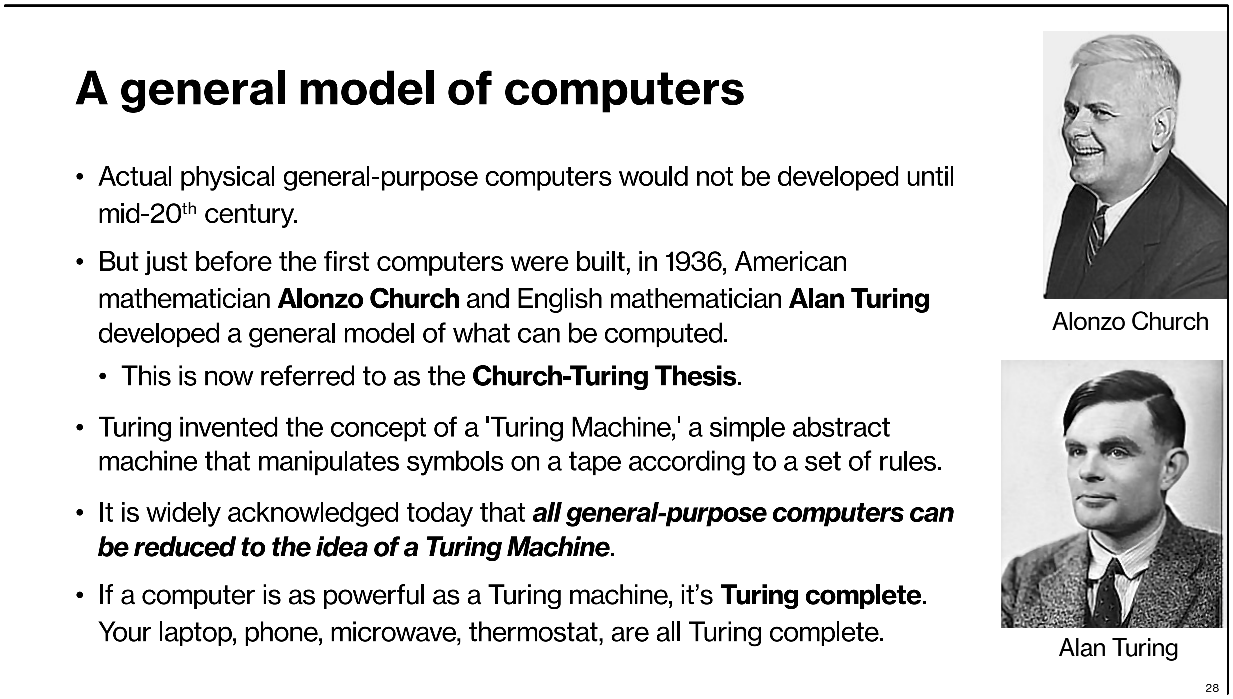
- In the late 1880s, American statistician **Herman Hollerith** built one of the world's first practical calculating machines, which he called a **tabulator**, to help compile census data.
- A U.S. census was taken every 10 years, but the growth of the population meant that it took 7.5 years to tally people by hand!
- Hollerith's tabulating machine tallied the entire census in only six weeks and completed the full analysis in just two and a half years.
- Hollerith set up the Tabulating Machine Company in 1896 which was renamed in 1924 as **IBM** (International Business Machines).

Herman Hollerith

tabulator

27

By the end of 19th century, engines of calculation as these devices were often known made their way into government and the public sector. This was motivated by the problem of conducting a US census every 10 years. The manual process was so arduous that it took 7.5 years to tally people by hand. Soon it would take more than 10 years to calculate the tally. American statistician Herman Hollerith (1860-1929) solved this problem by building the world's first practical machines, the tabulator, to tally census data in a matter of weeks and complete the full analysis in 2.5 years, saving the US government millions of dollars. Hollerith was so successful he went commercial. His company changed names and by 1924 it became what we now know as IBM, the International Business Machines company.



A general model of computers

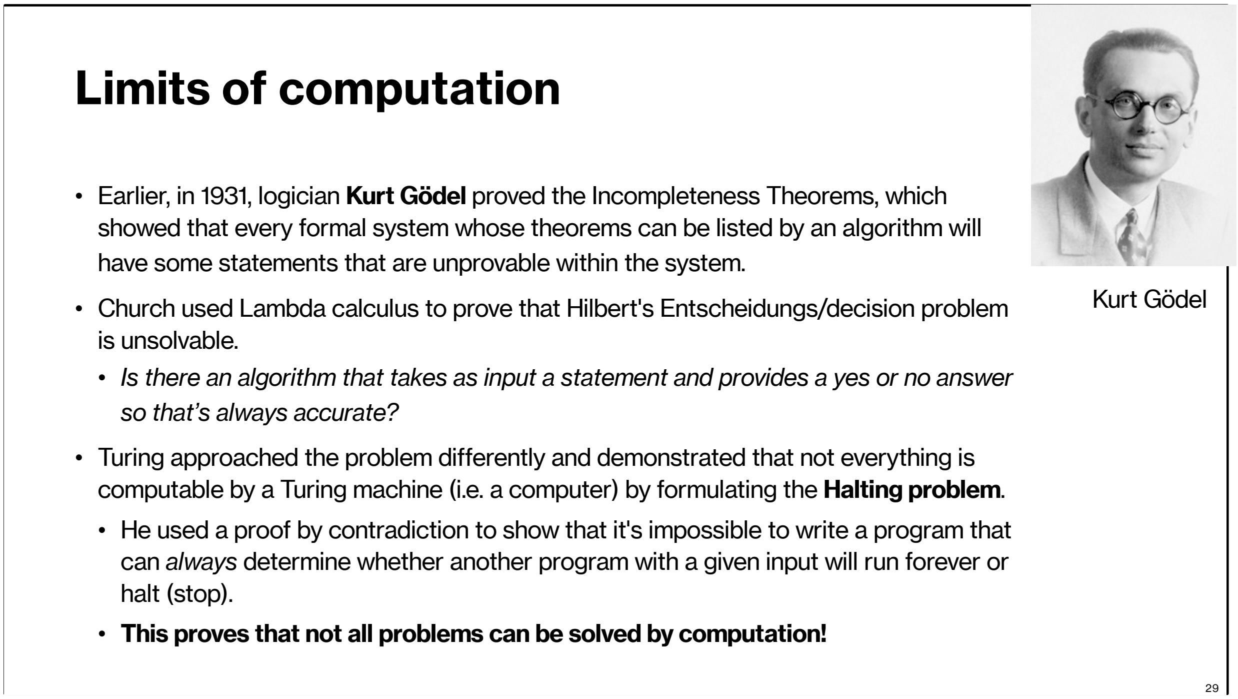
- Actual physical general-purpose computers would not be developed until mid-20th century.
- But just before the first computers were built, in 1936, American mathematician **Alonzo Church** and English mathematician **Alan Turing** developed a general model of what can be computed.
  - This is now referred to as the **Church-Turing Thesis**.
- Turing invented the concept of a 'Turing Machine,' a simple abstract machine that manipulates symbols on a tape according to a set of rules.
- It is widely acknowledged today that ***all general-purpose computers can be reduced to the idea of a Turing Machine***.
- If a computer is as powerful as a Turing machine, it’s **Turing complete**. Your laptop, phone, microwave, thermostat, are all Turing complete.

Alonzo Church

Alan Turing

28

As we will see, actual physical general-purpose computers would not be developed until mid-20th century. But just before the first computers were built, in 1936, American mathematician Alonzo Church (1903-1995) and English mathematician Alan Turing (1912-1954) developed a general model of what can be computed (by today's computers). This is now referred to as the Church-Turing Thesis. Turing invented the concept of a 'Turing Machine', a simple abstract machine that has wide-ranging capabilities. It is widely acknowledged today that all general-purpose computers can be reduced to the idea of a Turing Machine. This is why toward the end of this course we will learn about Turing machines. If a computer is as powerful as a Turing machine, it’s Turing complete. Your laptop, phone, microwave, thermostat, are all Turing complete. Alan Turing is one of the most important figures in computer science.



Limits of computation

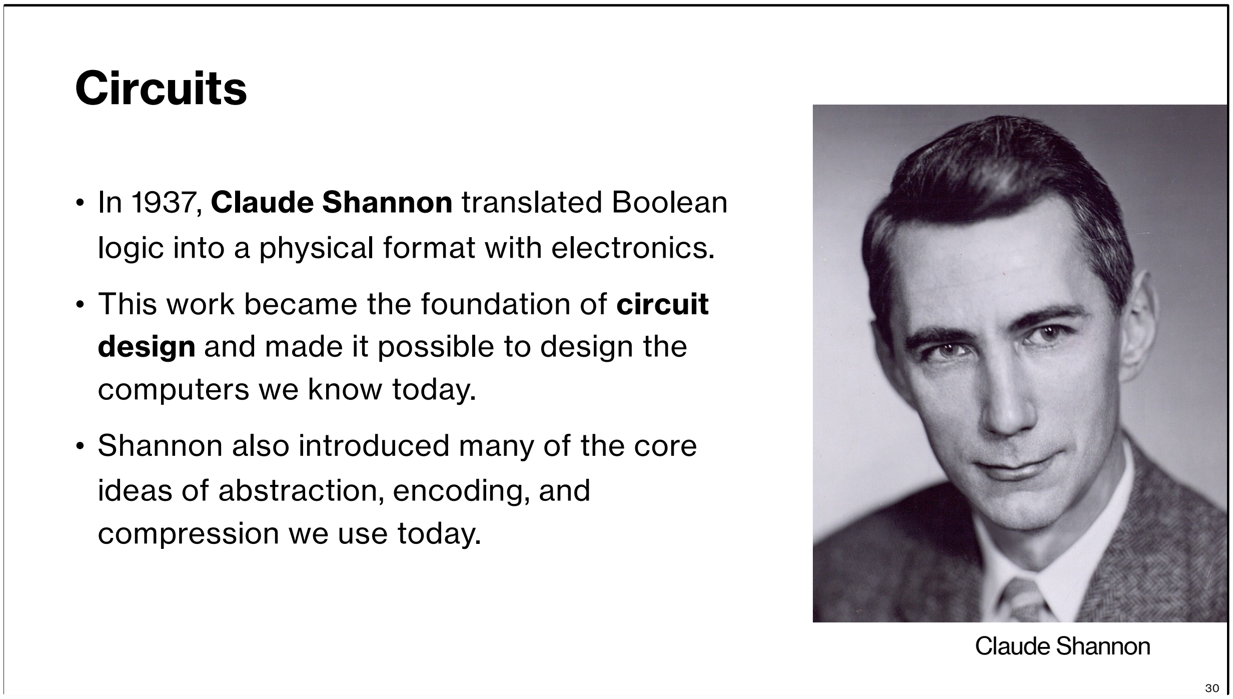
- Earlier, in 1931, logician **Kurt Gödel** proved the Incompleteness Theorems, which showed that every formal system whose theorems can be listed by an algorithm will have some statements that are unprovable within the system.
- Church used Lambda calculus to prove that Hilbert's Entscheidungs/decision problem is unsolvable.
  - *Is there an algorithm that takes as input a statement and provides a yes or no answer so that's always accurate?*
- Turing approached the problem differently and demonstrated that not everything is computable by a Turing machine (i.e. a computer) by formulating the **Halting problem**.
  - He used a proof by contradiction to show that it's impossible to write a program that can *always* determine whether another program with a given input will run forever or halt (stop).
  - **This proves that not all problems can be solved by computation!**

Kurt Gödel

29

The first half of the 20th century saw a lot of progress in establishing that computation has its limits. Earlier (in 1931), one of the most famous logicians to ever live, Kurt Gödel (1906-1978), proved the Incompleteness Theorems, which showed that every formal system whose theorems can be listed by an algorithm will have some statements that are unprovable within the system. Church proved that Hilbert's (German mathematician, 1862-1943) Entscheidungsproblem (which asks for an algorithm considers an inputted statement and answers "yes" or "no" according to whether it is universally valid, i.e., valid in every structure) is unsolvable. Turing demonstrated that not everything is computable by formulating the Halting problem. He used a proof by contradiction to show that it's impossible to write a program that can *always* determine whether another program with a given input will ever halt (stop). This proves that not all problems can be solved by computation! The general definition of what can be computed can be stated using the Church-Turing Thesis through the concept of a Turing Machine, which can compute anything a digital computer can compute.

You will learn more about those problems in CS101.

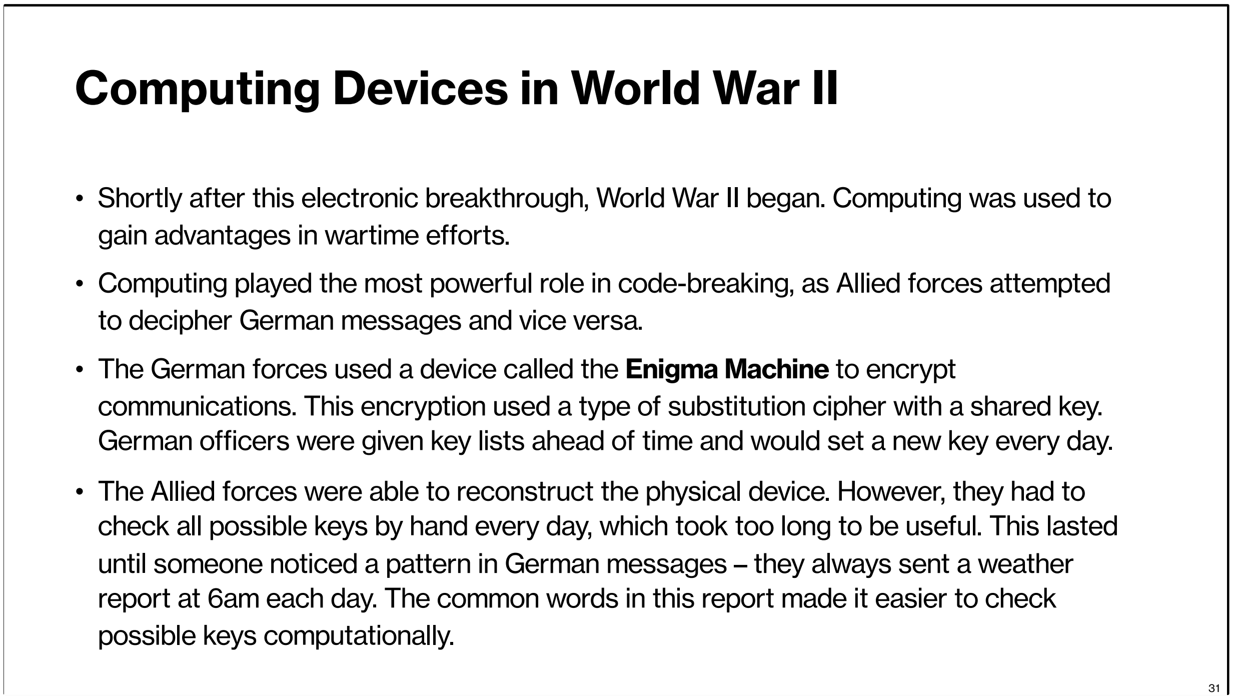


Circuits

- In 1937, **Claude Shannon** translated Boolean logic into a physical format with electronics.
- This work became the foundation of **circuit design** and made it possible to design the computers we know today.
- Shannon also introduced many of the core ideas of abstraction, encoding, and compression we use today.

Claude Shannon

In 1937, Claude Shannon (1916-2001) published his famous Master's Thesis, "A Symbolic Analysis of Relay and Switching Circuits". This was the first time Boolean logic was translated into a physical format with electronics. This work became the foundation of circuit design and made it possible to design the computers we know today (we will see examples of such circuits in a few weeks). Shannon also introduced many of the core ideas of abstraction, encoding, and compression we use today and he is considered the father of information theory.



Computing Devices in World War II

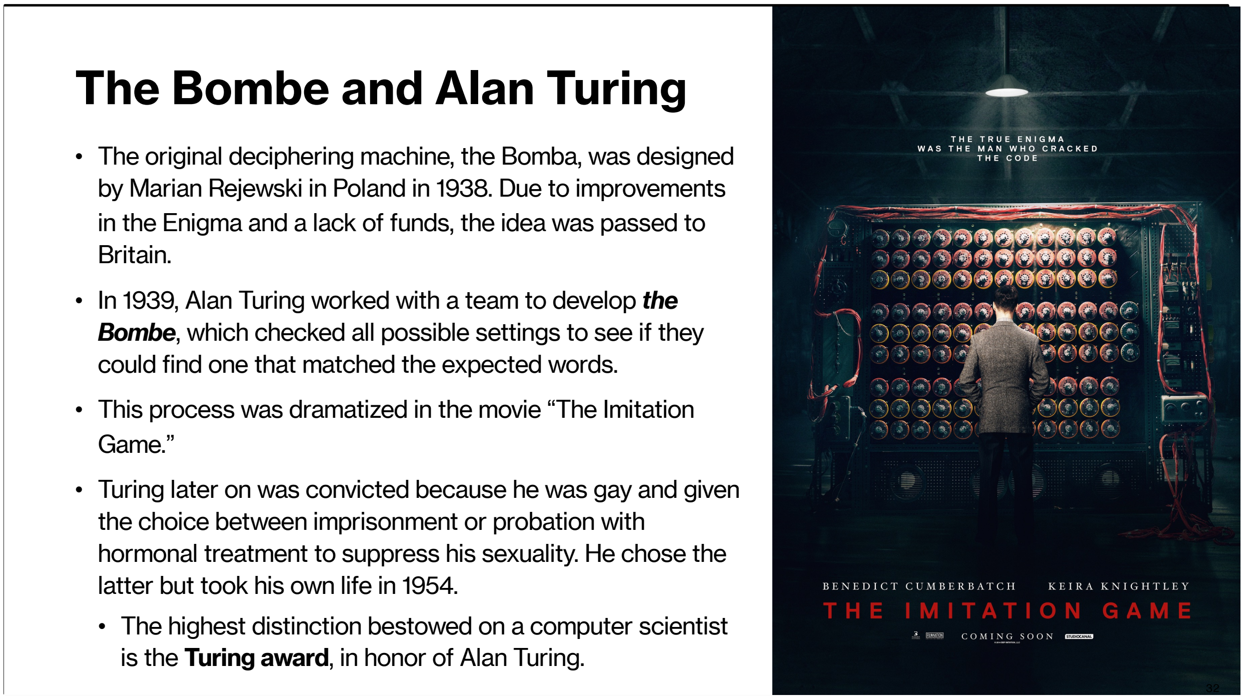
- Shortly after this electronic breakthrough, World War II began. Computing was used to gain advantages in wartime efforts.
- Computing played the most powerful role in code-breaking, as Allied forces attempted to decipher German messages and vice versa.
- The German forces used a device called the **Enigma Machine** to encrypt communications. This encryption used a type of substitution cipher with a shared key. German officers were given key lists ahead of time and would set a new key every day.
- The Allied forces were able to reconstruct the physical device. However, they had to check all possible keys by hand every day, which took too long to be useful. This lasted until someone noticed a pattern in German messages – they always sent a weather report at 6am each day. The common words in this report made it easier to check possible keys computationally.

Shortly after this electronic breakthrough, World War II began. Computing was used to gain advantages in wartime efforts.

Computing played the most powerful role in code-breaking, as Allied forces attempted to decipher German messages (and vice versa).

The German forces used a device called the Enigma Machine to encrypt communications. This encryption used a type of substitution cipher with a shared key. German officers were given key lists ahead of time and would set a new key every day.

The Allied forces were able to reconstruct the physical device. However, they had to check all possible keys by hand every day, which took too long to be useful. This lasted until someone noticed a pattern in German messages – they always sent a weather report at 6am each day. The common words in this report made it easier to check possible keys computationally.



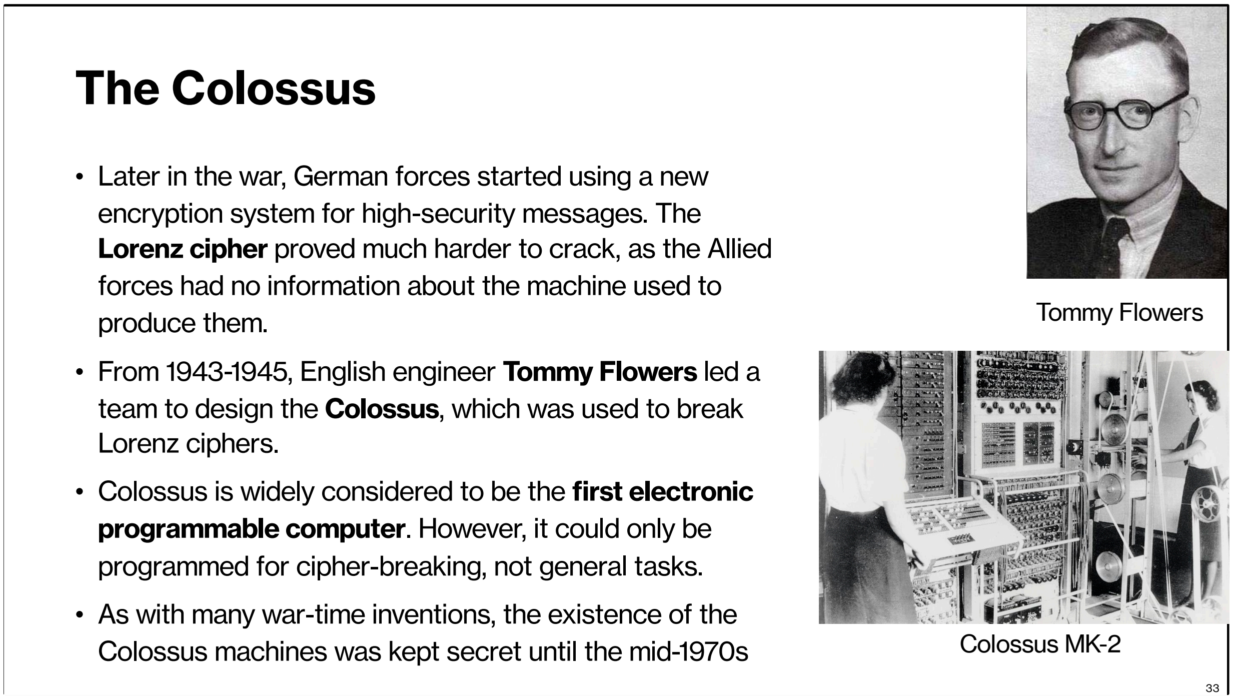
The Bombe and Alan Turing

- The original deciphering machine, the Bomba, was designed by Marian Rejewski in Poland in 1938. Due to improvements in the Enigma and a lack of funds, the idea was passed to Britain.
- In 1939, Alan Turing worked with a team to develop ***the Bombe***, which checked all possible settings to see if they could find one that matched the expected words.
- This process was dramatized in the movie “The Imitation Game.”
- Turing later on was convicted because he was gay and given the choice between imprisonment or probation with hormonal treatment to suppress his sexuality. He chose the latter but took his own life in 1954.
  - The highest distinction bestowed on a computer scientist is the **Turing award**, in honor of Alan Turing.

The original deciphering machine, the Bomba, was designed by Marian Rejewski in Poland in 1938. Due to improvements in the Enigma and a lack of funds, the idea was passed to Britain.

In 1939, Alan Turing worked with a team to develop *the Bombe*, which checked all possible settings to see if they could find one that matched the expected words. This process was dramatized in the movie “The Imitation Game” with Benedict Cumberbatch and Keira Knightley. Turing later on was convicted because he was gay and given the choice between imprisonment or probation with hormonal treatment to suppress his sexuality. He chose the latter but took his own life in 1954.

The highest distinction bestowed on a computer scientist is the Turing award, in honor of Alan Turing.



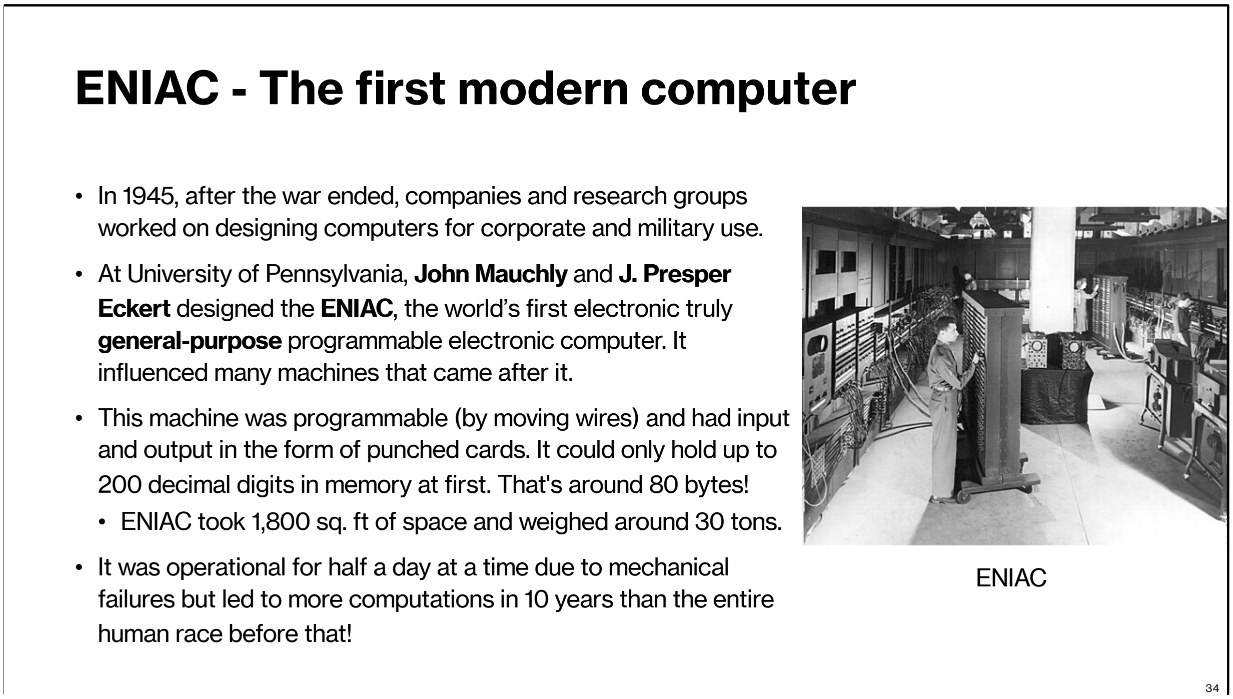
The Colossus

- Later in the war, German forces started using a new encryption system for high-security messages. The **Lorenz cipher** proved much harder to crack, as the Allied forces had no information about the machine used to produce them.
- From 1943-1945, English engineer **Tommy Flowers** led a team to design the **Colossus**, which was used to break Lorenz ciphers.
- Colossus is widely considered to be the **first electronic programmable computer**. However, it could only be programmed for cipher-breaking, not general tasks.
- As with many war-time inventions, the existence of the Colossus machines was kept secret until the mid-1970s

Tommy Flowers

Colossus MK-2

Later in the war, German forces started using a new encryption system for high-security messages. The Lorenz cipher proved much harder to crack, as the Allied forces had no information about the machine used to produce them. From 1943-1945, English engineer Tommy Flowers (1905-1998) led a team to design the Colossus, which was used to break Lorenz ciphers. Colossus is widely considered to be the first electronic programmable computer. However, it could only be programmed for cipher-breaking, not general tasks. As with many war-time inventions, the existence of the Colossus machines was kept secret until the mid-1970s.

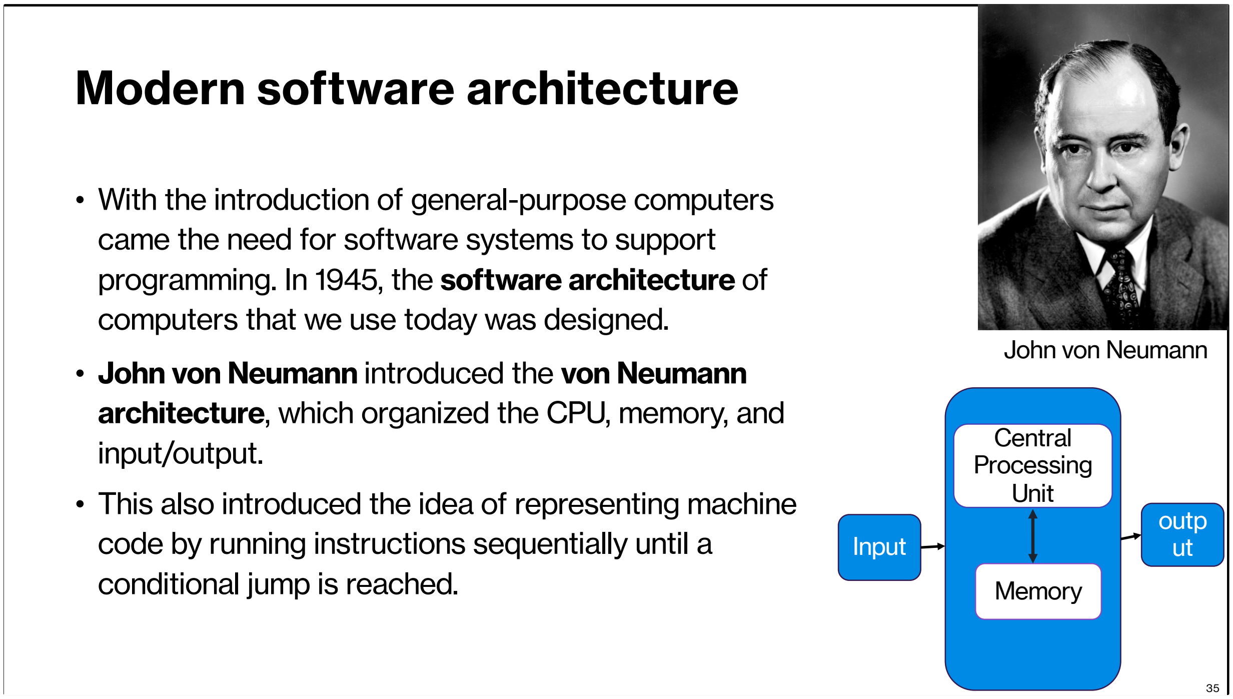


ENIAC - The first modern computer

- In 1945, after the war ended, companies and research groups worked on designing computers for corporate and military use.
- At University of Pennsylvania, **John Mauchly** and **J. Presper Eckert** designed the **ENIAC**, the world's first electronic truly **general-purpose** programmable electronic computer. It influenced many machines that came after it.
- This machine was programmable (by moving wires) and had input and output in the form of punched cards. It could only hold up to 200 decimal digits in memory at first. That's around 80 bytes!
  - ENIAC took 1,800 sq. ft of space and weighed around 30 tons.
- It was operational for half a day at a time due to mechanical failures but led to more computations in 10 years than the entire human race before that!

ENIAC

In 1945, after the war ended, companies and research groups started work on designing computers for corporate and military use.
At University of Pennsylvania, John Mauchly (1907-1980) and J. Presper Eckert (1919-1995) designed the ENIAC (Electronic Numerical Integrator and Computer), the world's first electronic truly general-purpose computer. It influenced many machines that came after it.
This machine was programmable (by moving wires) and had input and output in the form of punch cards. It could only hold up to 200 decimal digits in memory at first. That's only 80 bytes! It was operational for half a day at a time due to mechanical failures but led to more computations in 10 years than the entire human race before that!



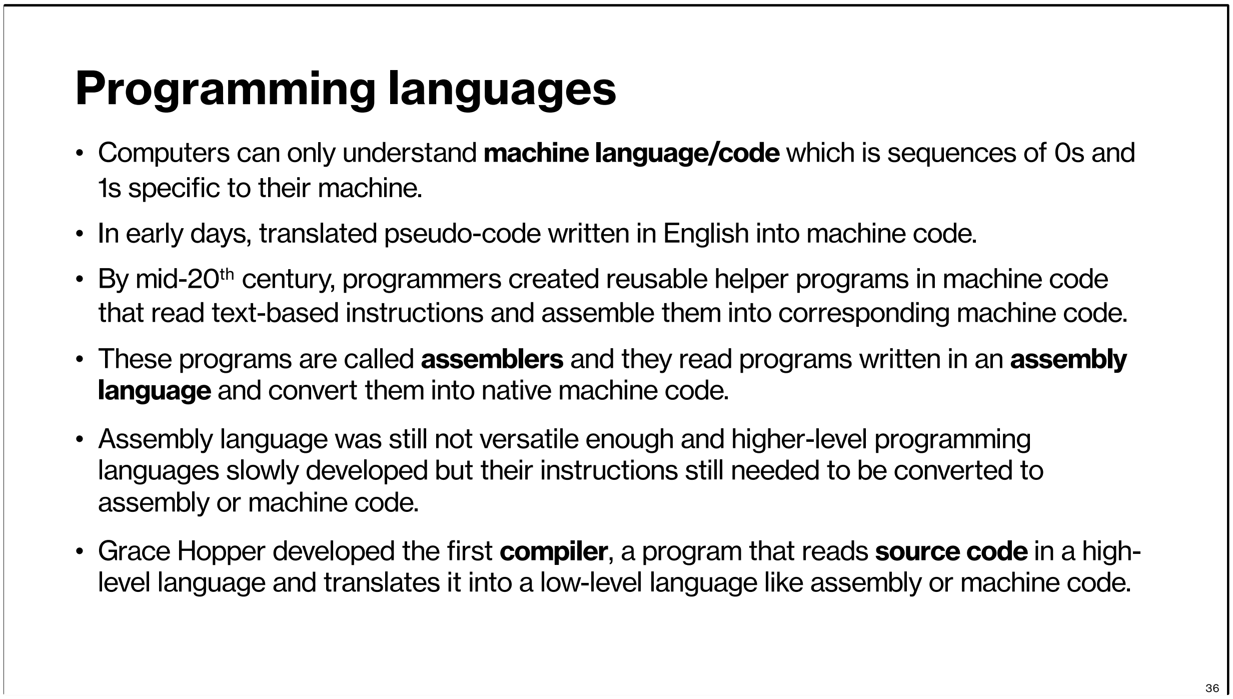
Modern software architecture

- With the introduction of general-purpose computers came the need for software systems to support programming. In 1945, the **software architecture** of computers that we use today was designed.
- **John von Neumann** introduced the **von Neumann architecture**, which organized the CPU, memory, and input/output.
- This also introduced the idea of representing machine code by running instructions sequentially until a conditional jump is reached.

John von Neumann

With the introduction of general-purpose computers came the need for software systems to support programming. Still in 1945, the software architecture of computers that we use today was designed. Hungarian-American mathematician, physicist, and computer scientists/engineer, John von Neumann introduced the von Neumann architecture, which organized the CPU, memory, and input/output.

This also introduced the idea of representing machine code by running instructions sequentially until a conditional jump is reached.



Programming languages

- Computers can only understand **machine language/code** which is sequences of 0s and 1s specific to their machine.
- In early days, translated pseudo-code written in English into machine code.
- By mid-20th century, programmers created reusable helper programs in machine code that read text-based instructions and assemble them into corresponding machine code.
- These programs are called **assemblers** and they read programs written in an **assembly language** and convert them into native machine code.
- Assembly language was still not versatile enough and higher-level programming languages slowly developed but their instructions still needed to be converted to assembly or machine code.
- Grace Hopper developed the first **compiler**, a program that reads **source code** in a high-level language and translates it into a low-level language like assembly or machine code.

Computers can only understand **machine language/code** which is sequences of 0s and 1s.
In early days, people had to write entire programs in machine code. They often started writing a high-level sketch of what the program should do on paper in English what we call a **pseudo-code**. Then they would expand it little by little to machine code.
By mid-20th century, programmers had developed slightly higher-level languages that were more human-readable and easier for programmers to remember and write correctly.

To do so, they created reusable helper programs in machine code that read text-based instructions and assemble them into corresponding machine code. These programs are called **assemblers** and they read programs written in **assembly language** and converts into native machine code.

Assembly language is still not versatile enough and higher programming languages slowly developed but their instructions still needed to be converted to assembly or machine code.

Hopper (1906-1992), American computer scientist, mathematician, and United States Navy rear admiral, developed the first **compiler**, a program that reads **source code** in a high-level language and translates it into a low-level language like assembly or machine code.

FORTRAN (Formula Translation) was released by IBM in 1957 and dominated early programming.

Computers can only understand **machine language/code** which is sequences of 0s and 1s specific to their machine.

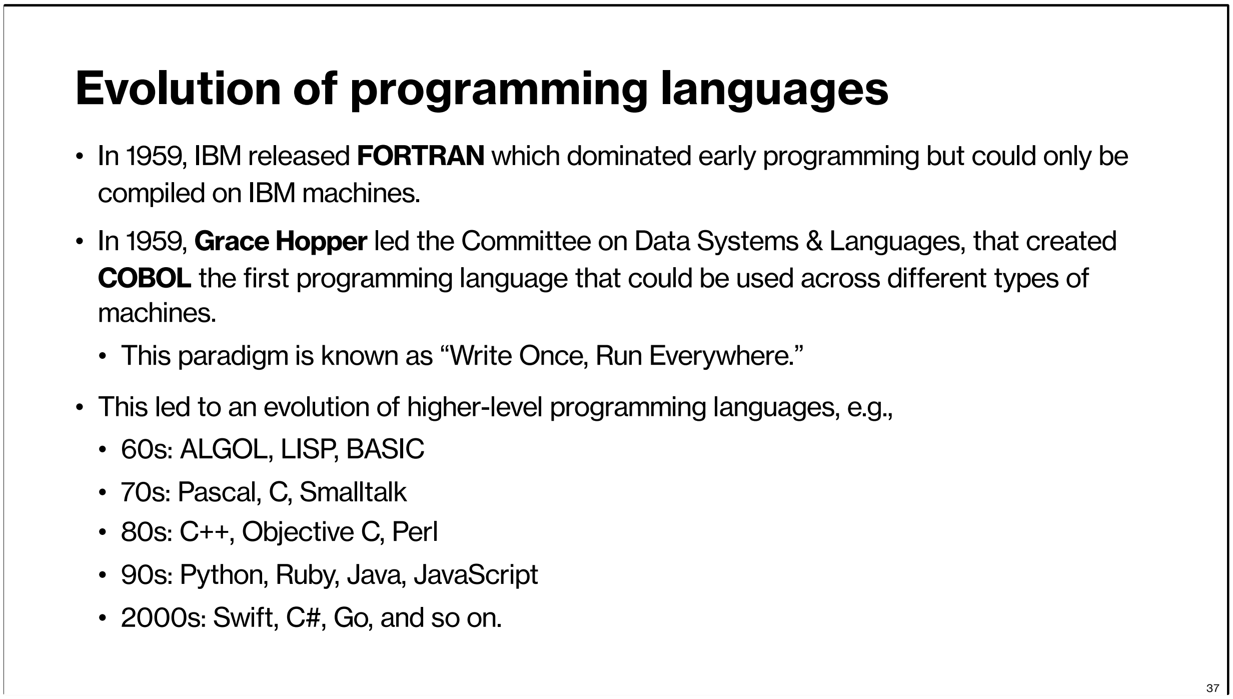
In early days, programmers started writing a high-level sketch of the program on paper in English in what we call **pseudo-code**. Then they would expand it little by little to machine code.

By mid-20th century, programmers created reusable helper programs in machine code that read text-based instructions and assemble them into corresponding machine code.

These programs are called **assemblers** and they read programs written in **assembly language** and convert them into native machine code.

Assembly language is still not versatile enough and higher-level programming languages slowly developed but their instructions still needed to be converted to assembly or machine code.

Grace Hopper developed the first **compiler**, a program that reads **source code** in a high-level language and translates it into a low-level language like assembly or machine code.



Evolution of programming languages

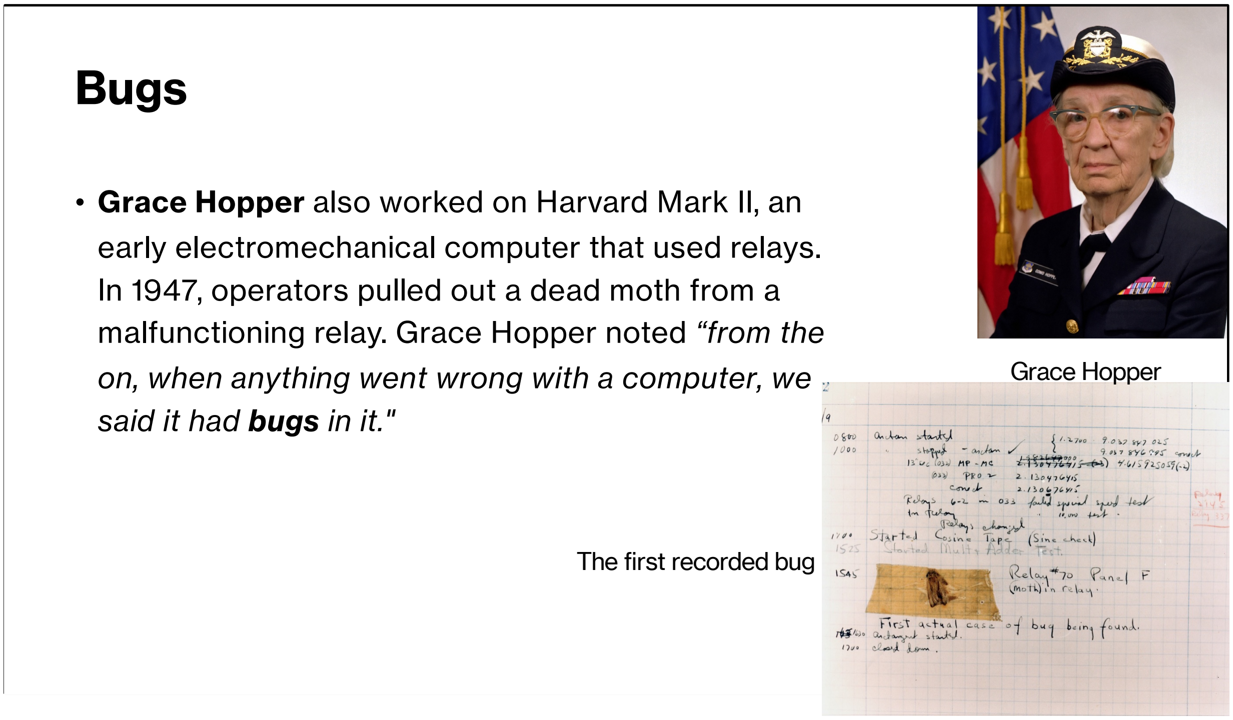
- In 1959, IBM released **FORTRAN** which dominated early programming but could only be compiled on IBM machines.
- In 1959, **Grace Hopper** led the Committee on Data Systems & Languages, that created **COBOL** the first programming language that could be used across different types of machines.
  - This paradigm is known as “Write Once, Run Everywhere.”
- This led to an evolution of higher-level programming languages, e.g.,
  - 60s: ALGOL, LISP, BASIC
  - 70s: Pascal, C, Smalltalk
  - 80s: C++, Objective C, Perl
  - 90s: Python, Ruby, Java, JavaScript
  - 2000s: Swift, C#, Go, and so on.

In 1959, IBM released **FORTRAN** (Formula Translation) which dominated early programming but could only be compiled on IBM machines.

In 1959, Grace Hopper led the Committee on Data Systems & Languages, that created **COBOL** (Common Business-Oriented Language), the first programming language that could be used across different types of machines.

This paradigm is known as “Write Once, Run Everywhere”.

This led to an evolution of higher-level programming languages, e.g., 60s saw ALGOL, LISP, BASIC, 70s had Pascal, C, Smalltalk, 80s brought C++, Objective C, Perl, 90s: Python, Ruby, Java, JavaScript, 2000s, Swift, C#, Go, and so on. Each new language uses new abstractions and technology to make programming faster and easier. Maybe eventually we will program in English?



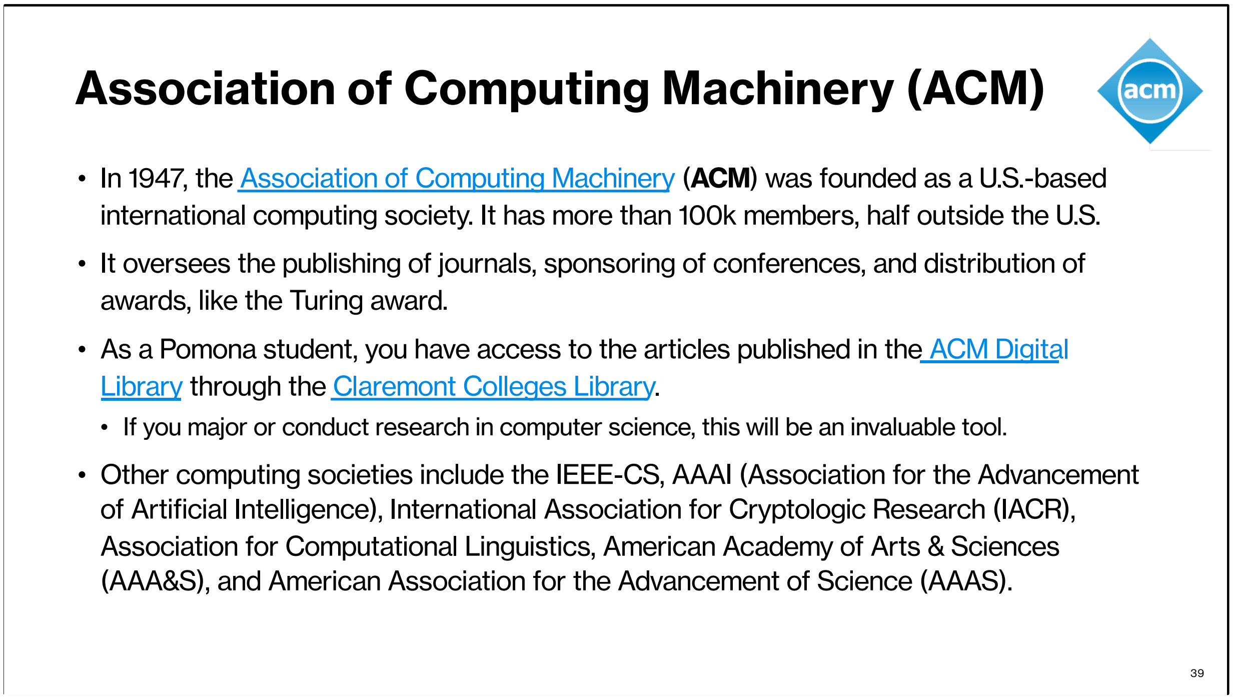
Bugs

- **Grace Hopper** also worked on Harvard Mark II, an early electromechanical computer that used relays. In 1947, operators pulled out a dead moth from a malfunctioning relay. Grace Hopper noted *“from the on, when anything went wrong with a computer, we said it had **bugs** in it."*

Grace Hopper

The first recorded bug

Grace Hopper (1906 –1992), an American computer scientist, mathematician, and United States Navy rear admiral, also worked on Harvard Mark II, an early electromechanical computer that used relays, electrically operated switches. In 1947, operators pulled out a dead moth from a malfunctioning relay and the term bug was since established to refer to errors in computer code.

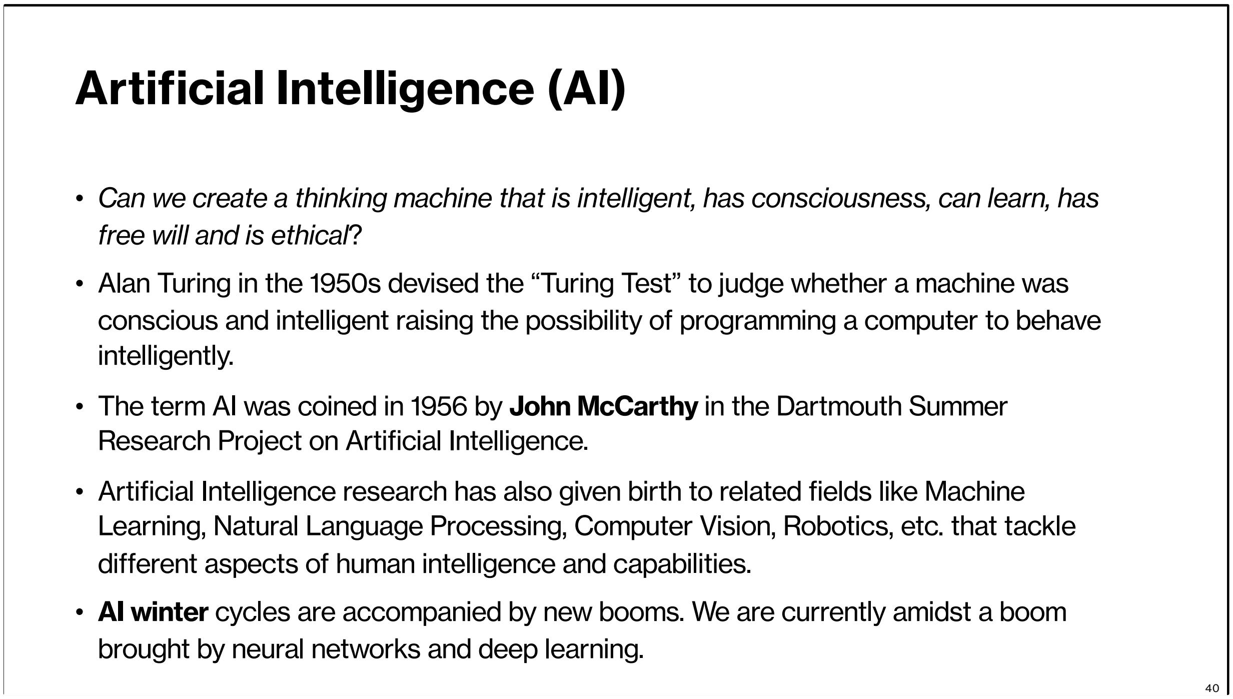


Association of Computing Machinery (ACM)

- In 1947, the Association of Computing Machinery (**ACM**) was founded as a U.S.-based international computing society. It has more than 100k members, half outside the U.S.
- It oversees the publishing of journals, sponsoring of conferences, and distribution of awards, like the Turing award.
- As a Pomona student, you have access to the articles published in the ACM Digital Library through the Claremont Colleges Library.
  - If you major or conduct research in computer science, this will be an invaluable tool.
- Other computing societies include the IEEE-CS, AAAI (Association for the Advancement of Artificial Intelligence), International Association for Cryptologic Research (IACR), Association for Computational Linguistics, American Academy of Arts & Sciences (AAA&S), and American Association for the Advancement of Science (AAAS).

39

As computers became more widespread, a new international society called Association of Computing Machinery, or ACM, was formed mid century. ACM still exists and has more than 100k members throughout the world. One of the main characteristics is that it has local chapters and special interest groups (SIGs) that oversee the publication of journals, sponsorship of conferences, and distribution of awards like the Turing award. As a 5C student you have access to articles published in the ACM DL through the Claremont Colleges Library. If you ever work on cs research or major in cs, you will get to use the ACM DL. There are other famous computing societies, including IEEE-CS, AAAI (Association for the Advancement of Artificial Intelligence), International Association for Cryptologic Research (IACR), Association for Computational Linguistics, American Academy of Arts & Sciences (AAA&S), and American Association for the Advancement of Science (AAAS).



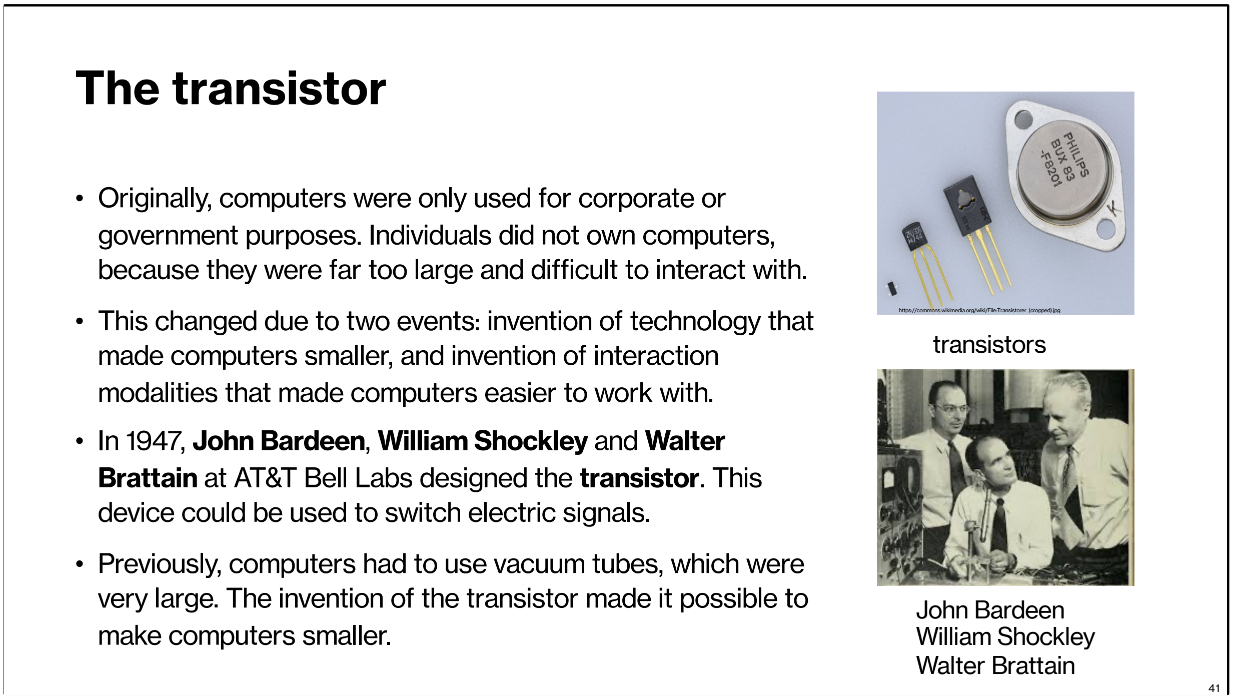
Artificial Intelligence (AI)

- *Can we create a thinking machine that is intelligent, has consciousness, can learn, has free will and is ethical*?
- Alan Turing in the 1950s devised the “Turing Test” to judge whether a machine was conscious and intelligent raising the possibility of programming a computer to behave intelligently.
- The term AI was coined in 1956 by **John McCarthy** in the Dartmouth Summer Research Project on Artificial Intelligence.
- Artificial Intelligence research has also given birth to related fields like Machine Learning, Natural Language Processing, Computer Vision, Robotics, etc. that tackle different aspects of human intelligence and capabilities.
- **AI winter** cycles are accompanied by new booms. We are currently amidst a boom brought by neural networks and deep learning.

Mid-century also saw the rise of interest in AI with the question of can we create a thinking machine that is intelligent, has consciousness, can learn, has free will and is ethical? fascinating computer scientists. The philosophical roots of AI go back centuries with Rene Descartes arguing that a powerful evil demon or mad scientist could exist who sets out to manipulate and deceive subjects, thereby preventing from knowing the true nature of reality. Alan Turing in the 1950s devised the “Turing Test” to judge whether a machine was conscious and intelligent raising the possibility of programming a computer to behave intelligently.

The term AI was coined in 1956 by John McCarthy in the Dartmouth Summer Research Project on Artificial Intelligence.

Artificial Intelligence research has also given birth to related fields like Machine Learning, Natural Language Processing, Computer Vision, Robotics, etc that tackle different aspects of human intelligence and capabilities. Across the past century, AI cycles between periods of decreased interest and disillusionment on AI’s capabilities known as AI winters, followed by new booms due to new developments. We are currently amidst a boom brought by neural networks and deep learning that has resulted into ChatGPT and the rest of the large language models.



The transistor

- Originally, computers were only used for corporate or government purposes. Individuals did not own computers, because they were far too large and difficult to interact with.
- This changed due to two events: invention of technology that made computers smaller, and invention of interaction modalities that made computers easier to work with.
- In 1947, **John Bardeen**, **William Shockley** and **Walter Brattain** at AT&T Bell Labs designed the **transistor**. This device could be used to switch electric signals.
- Previously, computers had to use vacuum tubes, which were very large. The invention of the transistor made it possible to make computers smaller.

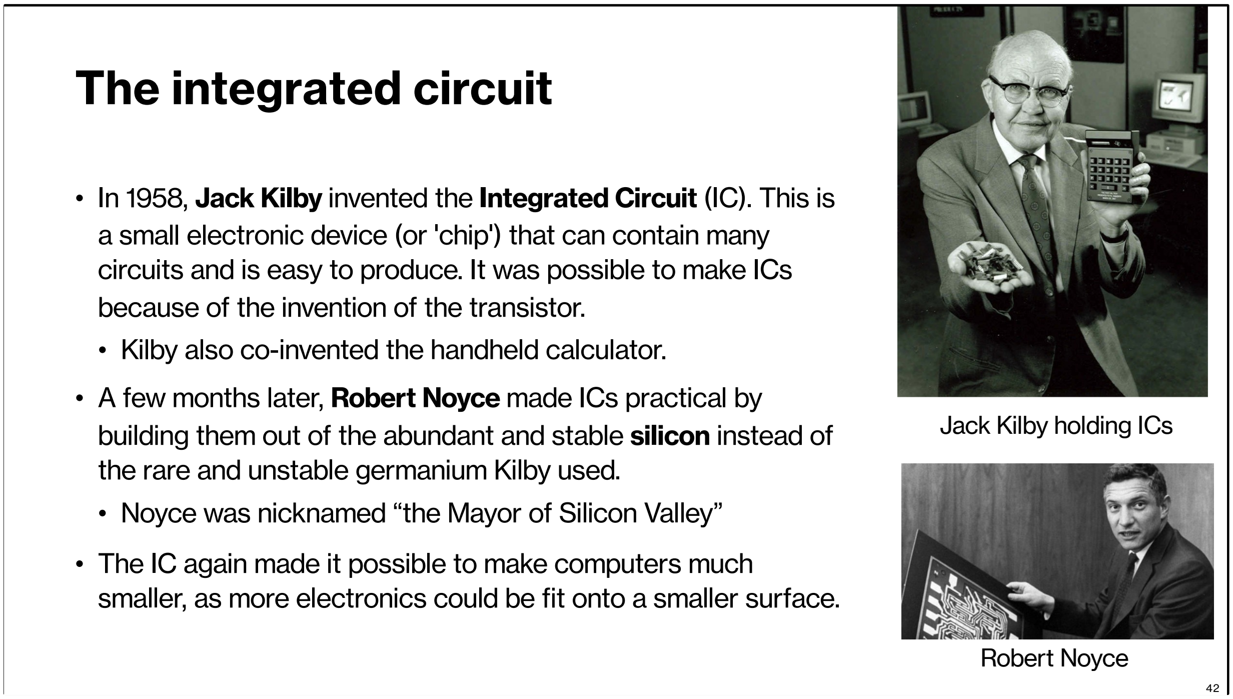
https://commons.wikimedia.org/wiki/File:Transistorer_(cropped).jpg

transistors

John Bardeen
William Shockley
Walter Brattain

41

Originally, computers were only used for corporate or government purposes. Individuals did not own computers, because they were far too large and difficult to interact with.
This changed due to two events: invention of technology that made computers smaller, and invention of interaction modalities that made computers easier to work with.
In 1947, John Bardeen (1908-1991), William Shockley (1910-1989) and Walter Brattain (1902-1987) at AT&T Bell Labs designed the transistor. This device could be used to switch electric signals.
Previously, computers had to use vacuum tubes, which were very large. The invention of the transistor made it possible to make computers smaller.



The integrated circuit

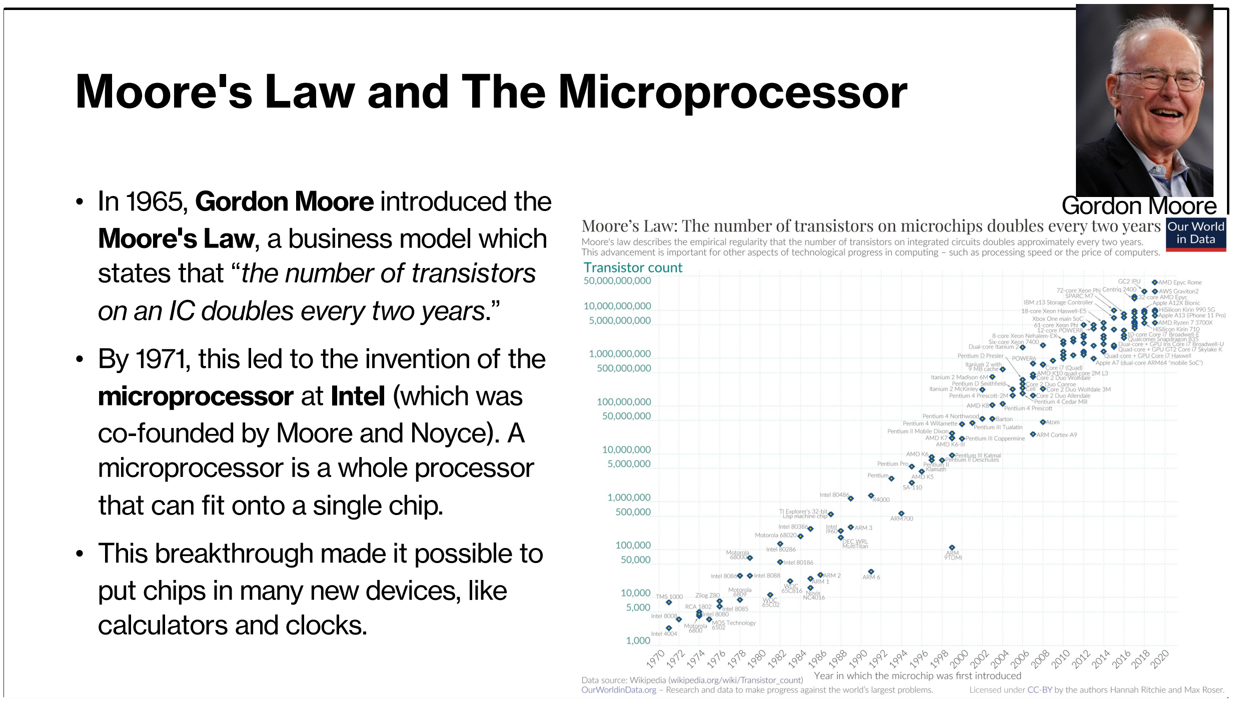
- In 1958, **Jack Kilby** invented the **Integrated Circuit** (IC). This is a small electronic device (or 'chip') that can contain many circuits and is easy to produce. It was possible to make ICs because of the invention of the transistor.
  - Kilby also co-invented the handheld calculator.
- A few months later, **Robert Noyce** made ICs practical by building them out of the abundant and stable **silicon** instead of the rare and unstable germanium Kilby used.
  - Noyce was nicknamed “the Mayor of Silicon Valley”
- The IC again made it possible to make computers much smaller, as more electronics could be fit onto a smaller surface.

Jack Kilby holding ICs

Robert Noyce

In 1958, American electrical engineer Jack Kilby (1923-2005) invented the Integrated Circuit (IC). This is a small electronic device (or 'chip') that can contain a large number of circuits and is easy to produce. It was possible to make ICs because of the invention of the transistor. A few months later, Robert Noyce (1927-1990) made ICs practical by building them out of the abundant silicon instead of the rare and unstable germanium. Noyce was nicknamed “the Mayor of Silicon Valley”

The IC again made it possible to make computers much smaller, as more electronics could be fit onto a smaller surface.



Moore's Law and The Microprocessor

- In 1965, **Gordon Moore** introduced the **Moore's Law**, a business model which states that "*the number of transistors on an IC doubles every two years.*"
- By 1971, this led to the invention of the **microprocessor** at **Intel** (which was co-founded by Moore and Noyce). A microprocessor is a whole processor that can fit onto a single chip.
- This breakthrough made it possible to put chips in many new devices, like calculators and clocks.

Gordon Moore

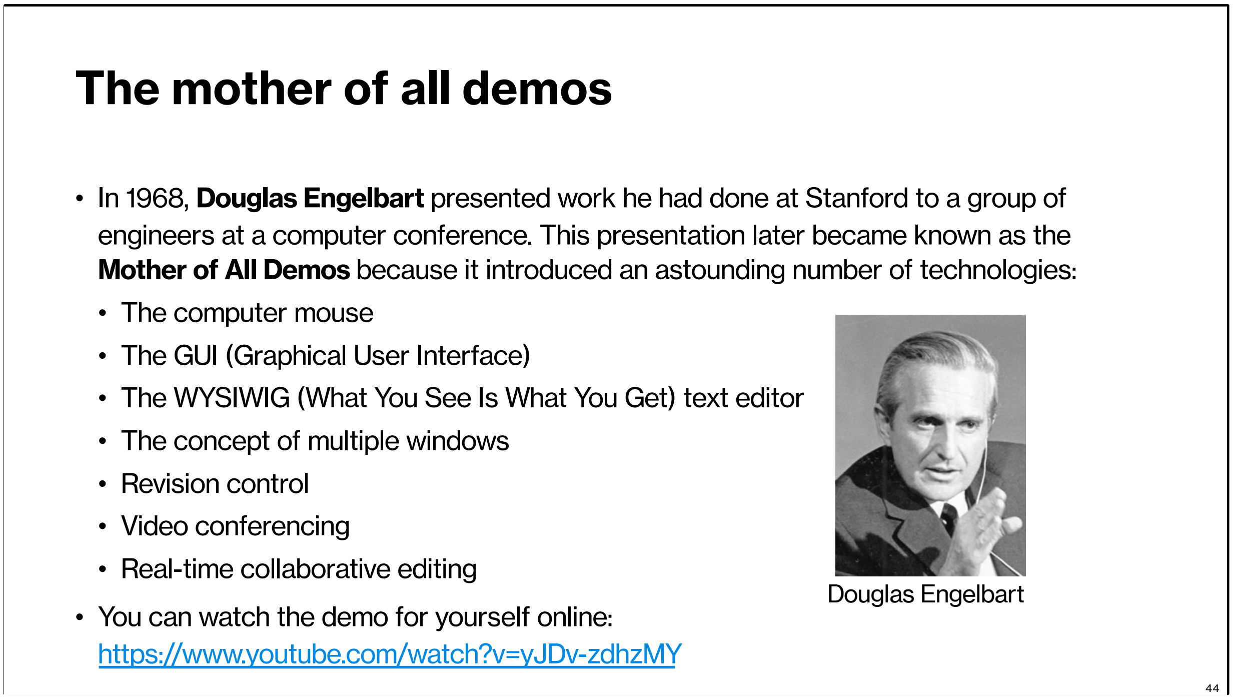
Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

In 1965, American businessman, engineer, and co-founder of Intel, Gordon Moore (1929-2023) introduced the business model known as Moore's Law, which states that the number of transistors on an IC will double every two years. By 1971, this led to the invention of the microprocessor at Intel. A microprocessor is a whole processor that can fit onto a single chip. This breakthrough made it possible to put chips in many new devices, like calculators and clocks.



The mother of all demos

- In 1968, **Douglas Engelbart** presented work he had done at Stanford to a group of engineers at a computer conference. This presentation later became known as the **Mother of All Demos** because it introduced an astounding number of technologies:
  - The computer mouse
  - The GUI (Graphical User Interface)
  - The WYSIWIG (What You See Is What You Get) text editor
  - The concept of multiple windows
  - Revision control
  - Video conferencing
  - Real-time collaborative editing
- You can watch the demo for yourself online: https://www.youtube.com/watch?v=yJDv-zdhzMY

Douglas Engelbart

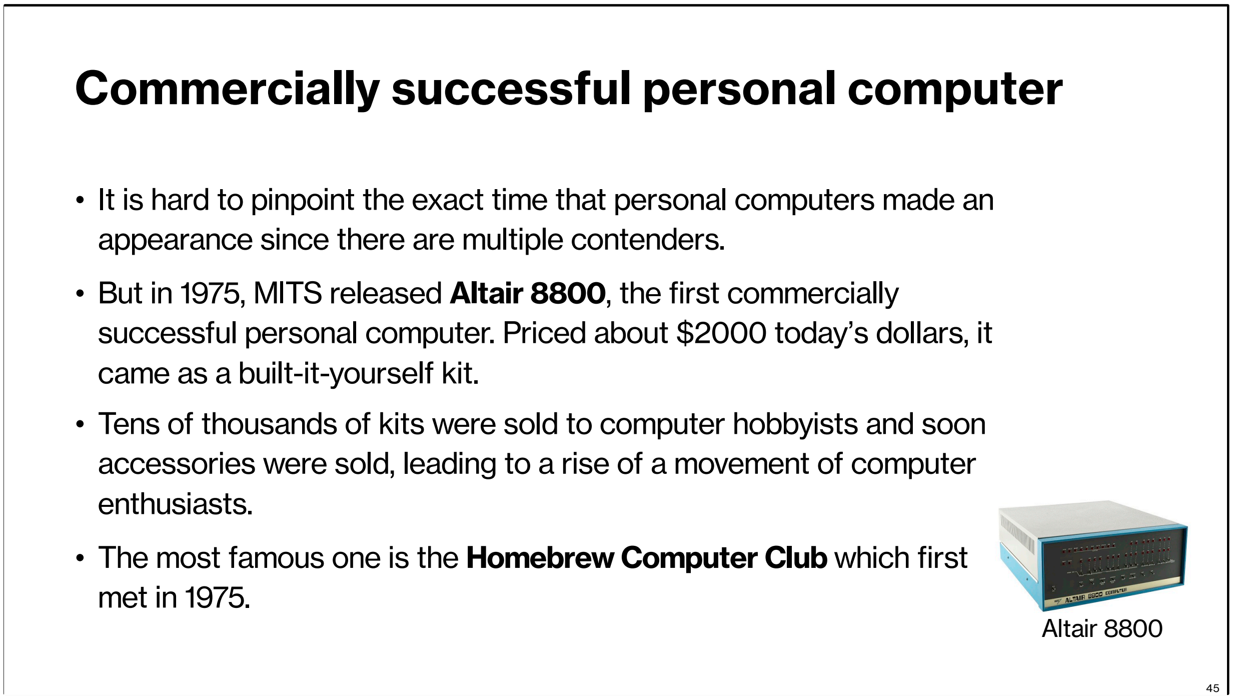
44

In 1968, Douglas Engelbart (1925-2013) presented work he had done at the Augmentation Research Center at Stanford to a group of engineers at a computer conference. This 1.5 hour presentation later became known as the Mother of All Demos because it introduced an astounding number of technologies that we use to this day.:

The computer mouse
The GUI (Graphical User Interface)
The WYSIWIG (What You See Is What You Get) text editor
The concept of multiple windows
Revision control

Video conferencing
Real-time collaborative editing

I HIGHLY recommend you watch for yourself online: https://www.youtube.com/watch?v=yJDv-zdhzMY

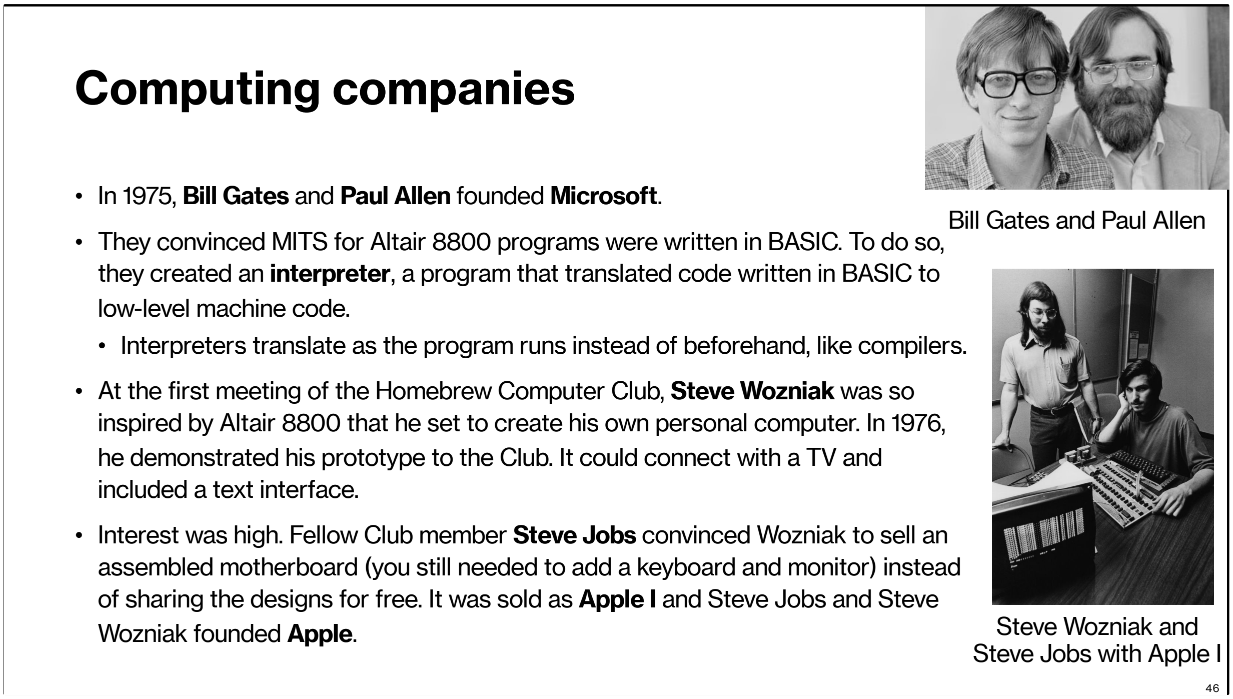


Commercially successful personal computer

- It is hard to pinpoint the exact time that personal computers made an appearance since there are multiple contenders.
- But in 1975, MITS released **Altair 8800**, the first commercially successful personal computer. Priced about $2000 today's dollars, it came as a built-it-yourself kit.
- Tens of thousands of kits were sold to computer hobbyists and soon accessories were sold, leading to a rise of a movement of computer enthusiasts.
- The most famous one is the **Homebrew Computer Club** which first met in 1975.

Altair 8800

It is hard to pinpoint the exact time that personal computers made an appearance since there are multiple contenders. But in 1975, the first commercially successful personal computer, **Altair 8800** was released. Priced about $2000 today's dollars (that was considered cheap for computers), it came as a built-it-yourself kit. Tens of thousands of kits were sold to computer hobbyists and soon accessories were sold, leading to a rise of a movement of computer enthusiasts. The most famous one is the **Homebrew Computer Club** which first met in 1975. We will come back to that in a minute.



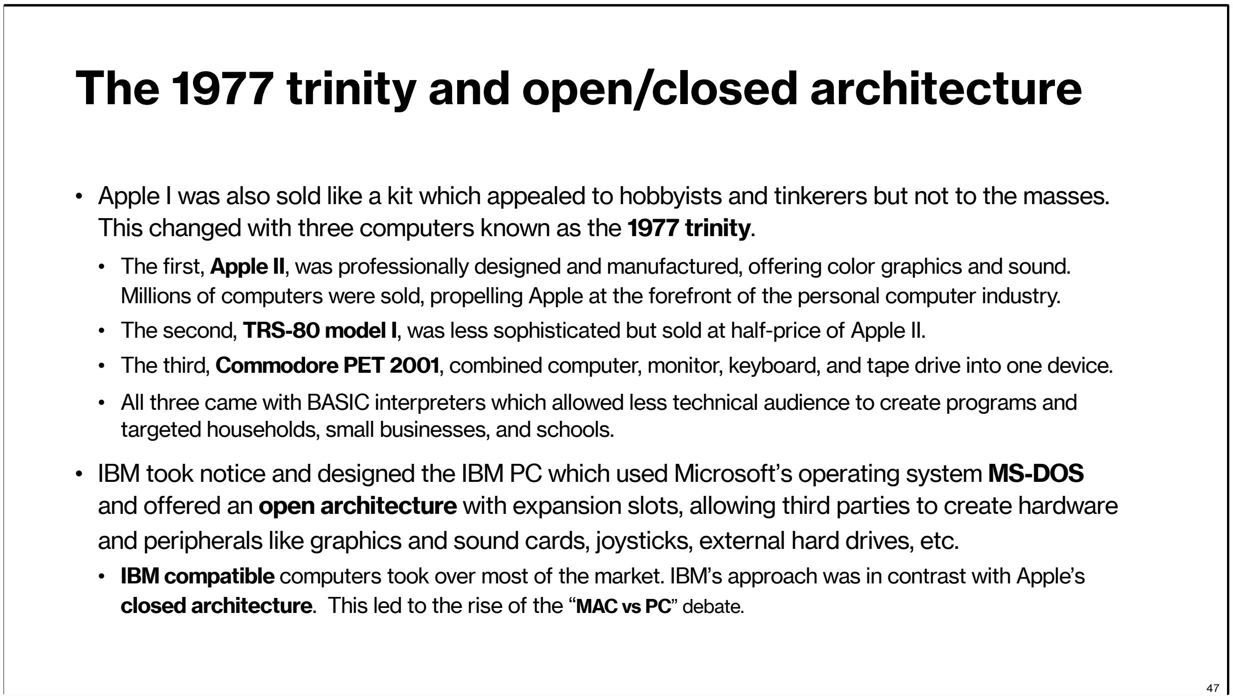
Computing companies

- In 1975, **Bill Gates** and **Paul Allen** founded **Microsoft**.
- They convinced MITS for Altair 8800 programs were written in BASIC. To do so, they created an **interpreter**, a program that translated code written in BASIC to low-level machine code.
  - Interpreters translate as the program runs instead of beforehand, like compilers.
- At the first meeting of the Homebrew Computer Club, **Steve Wozniak** was so inspired by Altair 8800 that he set to create his own personal computer. In 1976, he demonstrated his prototype to the Club. It could connect with a TV and included a text interface.
- Interest was high. Fellow Club member **Steve Jobs** convinced Wozniak to sell an assembled motherboard (you still needed to add a keyboard and monitor) instead of sharing the designs for free. It was sold as **Apple I** and Steve Jobs and Steve Wozniak founded **Apple**.

Bill Gates and Paul Allen

Steve Wozniak and Steve Jobs with Apple I

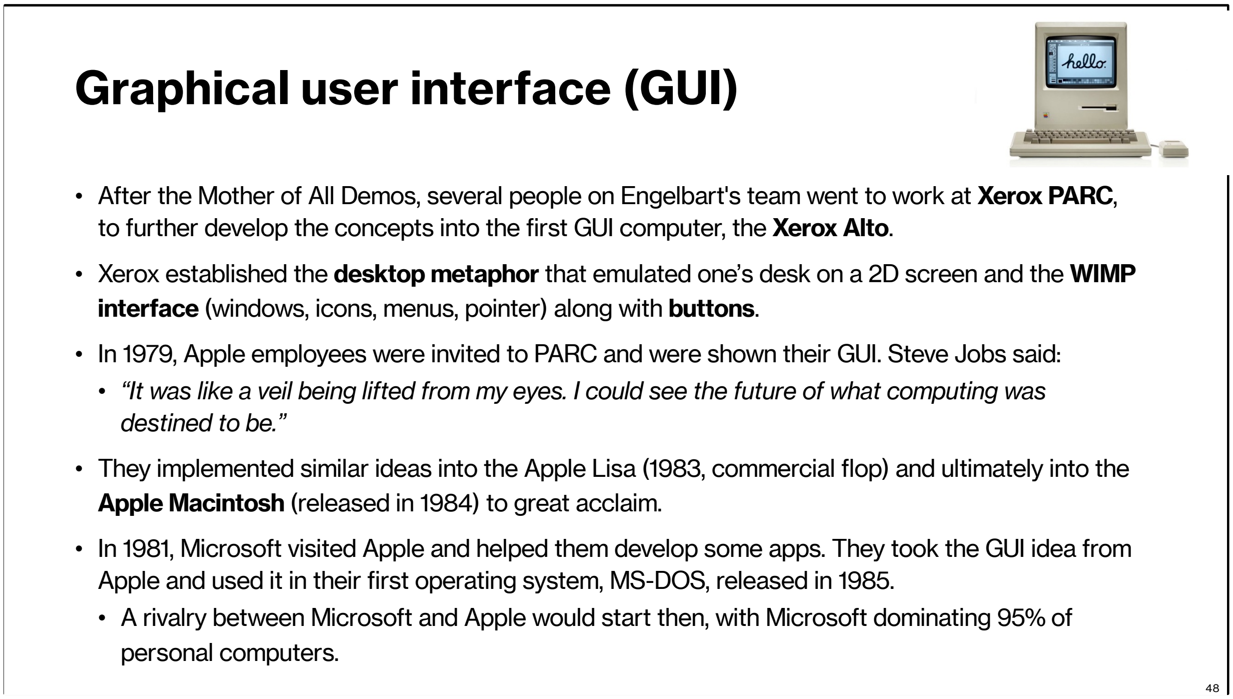
In 1975, Bill Gates (1955-) and Paul Allen (1953-2018) founded Microsoft. They convinced MITS, the maker of Altair 8800, that it would be better if the computer could run programs written in the programming language BASIC. To do so, they created an interpreter, a program that translated code written in BASIC to low-level machine code. Interpreters do the translation as the program runs instead of beforehand, as compilers do. At the first meeting of the Homebrew Computer Club, Steve Wozniak (1950-) was so inspired by Altair 8800 that he set to create his own personal computer. In 1976, he demonstrated his prototype to the Club. It could connect with a TV and included a text interface. Interest was high. Fellow Club member Steve Jobs (1955-2011) convinced Wozniak to sell an assembled motherboard (you still needed to add a keyboard and monitor) instead of sharing the designs for free. It was sold as Apple I and Steve Jobs and Steve Wozniak founded Apple.



The 1977 trinity and open/closed architecture

- Apple I was also sold like a kit which appealed to hobbyists and tinkerers but not to the masses. This changed with three computers known as the **1977 trinity**.
  - The first, **Apple II**, was professionally designed and manufactured, offering color graphics and sound. Millions of computers were sold, propelling Apple at the forefront of the personal computer industry.
  - The second, **TRS-80 model I**, was less sophisticated but sold at half-price of Apple II.
  - The third, **Commodore PET 2001**, combined computer, monitor, keyboard, and tape drive into one device.
  - All three came with BASIC interpreters which allowed less technical audience to create programs and targeted households, small businesses, and schools.
- IBM took notice and designed the IBM PC which used Microsoft's operating system **MS-DOS** and offered an **open architecture** with expansion slots, allowing third parties to create hardware and peripherals like graphics and sound cards, joysticks, external hard drives, etc.
  - **IBM compatible** computers took over most of the market. IBM's approach was in contrast with Apple's **closed architecture**. This led to the rise of the "**MAC vs PC**" debate.

Apple I was also sold like a kit which appealed to hobbyists and tinkerers but not to the masses. This changed with three computers known as the 1977 trinity. The first, Apple II, was professionally designed and manufactured, offering color graphics and sound. Millions of computers were sold, propelling Apple at the forefront of the personal computer industry. The second, TRS-80 model I, was less sophisticated but sold at half-price of Apple II and sold very well. The third, Commodore PET 2001, combined computer, monitor, keyboard, and tape drive into one device making it look like an appliance. All three came with BASIC interpreters which allowed less technical audience to create programs and targeted households, small businesses, and schools. IBM who was falling behind in the personal computer battles as its computers aimed governments and businesses and were massive and expensive took notice and designed the IBM PC which used Microsoft's operating system MS-DOS and offered an open architecture with expansion slots, allowing third parties to create hardware and peripherals like graphics and sound cards, joysticks, external hard drives, etc. IBM compatible computers took over most of the market as people liked the flexibility of being able to expand their PC's capabilities. IBM's approach was in contrast with Apple's closed architecture which did everything in-house. This led to the rise of the "MAC vs PC" debate, although all are personal computers.



Graphical user interface (GUI)

- After the Mother of All Demos, several people on Engelbart's team went to work at **Xerox PARC**, to further develop the concepts into the first GUI computer, the **Xerox Alto**.
- Xerox established the **desktop metaphor** that emulated one's desk on a 2D screen and the **WIMP interface** (windows, icons, menus, pointer) along with **buttons**.
- In 1979, Apple employees were invited to PARC and were shown their GUI. Steve Jobs said:
  - *"It was like a veil being lifted from my eyes. I could see the future of what computing was destined to be."*
- They implemented similar ideas into the Apple Lisa (1983, commercial flop) and ultimately into the **Apple Macintosh** (released in 1984) to great acclaim.
- In 1981, Microsoft visited Apple and helped them develop some apps. They took the GUI idea from Apple and used it in their first operating system, MS-DOS, released in 1985.
  - A rivalry between Microsoft and Apple would start then, with Microsoft dominating 95% of personal computers.

After the Mother of All Demos, several people on Engelbart's team went to work at Xerox PARC, to further develop the concepts into the first GUI computer, the Xerox Alto.

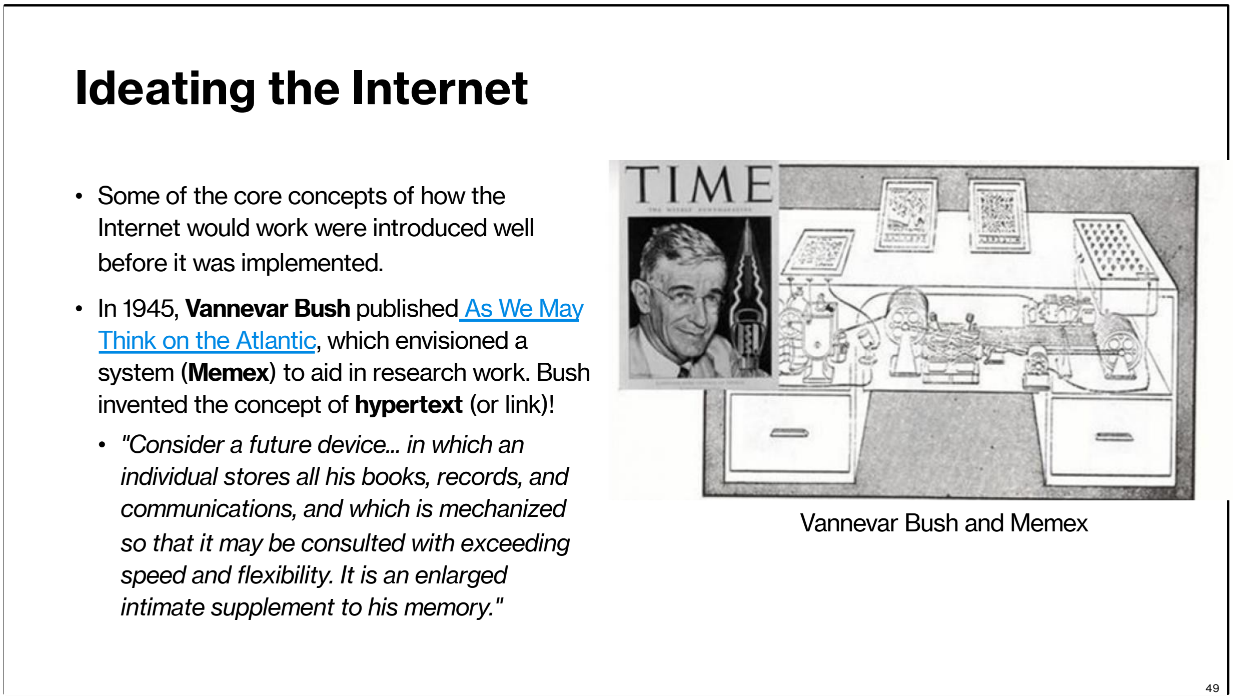
Xerox established the desktop metaphor that emulated one's desk on a 2D screen and the WIMP interface (windows, icons, menus, pointer) along with buttons.

In 1979, Apple employees were invited to PARC and were shown their GUI implementation. Steve Jobs said:

> *"It was like a veil being lifted from my eyes. I could see the future of what computing was destined to be."*

They implemented similar ideas into the Apple Lisa (1983, commercial flop) and ultimately into the Apple Macintosh (released in 1984) to great acclaim.
In 1981, Microsoft visited Apple and helped them develop some apps. They took the GUI idea from Apple and used it in their first operating system, MS-DOS, released in 1985.

A rivalry between Microsoft and Apple would start then, with Microsoft dominating 95% of personal computers.

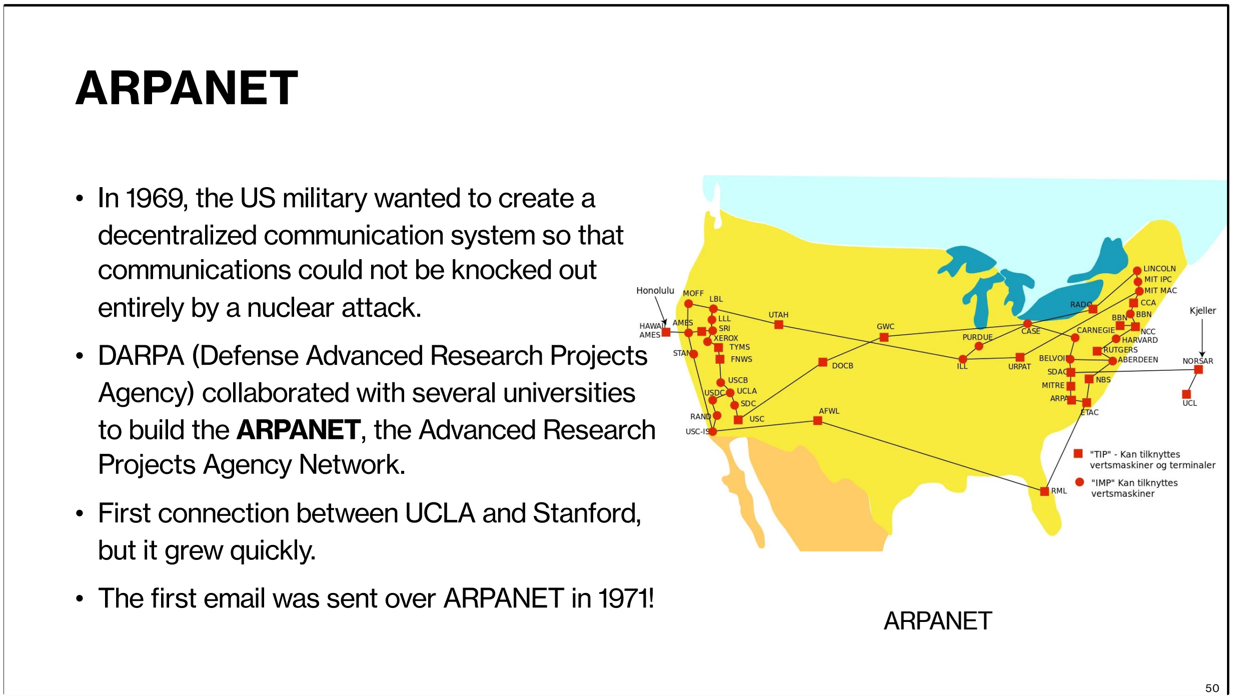


Ideating the Internet

- Some of the core concepts of how the Internet would work were introduced well before it was implemented.
- In 1945, **Vannevar Bush** published [As We May Think on the Atlantic](), which envisioned a system (**Memex**) to aid in research work. Bush invented the concept of **hypertext** (or link)!
  - *"Consider a future device... in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."*

Vannevar Bush and Memex

Some of the core concepts of how the Internet would work were introduced well before it was implemented. In 1945, American engineer and inventor Vannevar Bush (1890-1974) published the influential essay As We May Think on the Atlantic, which envisioned a system (Memex) to aid in research work. Bush invented the concept of hypertext (or link)! Here’s an interesting excerpt about Meex: *"Consider a future device... in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."*



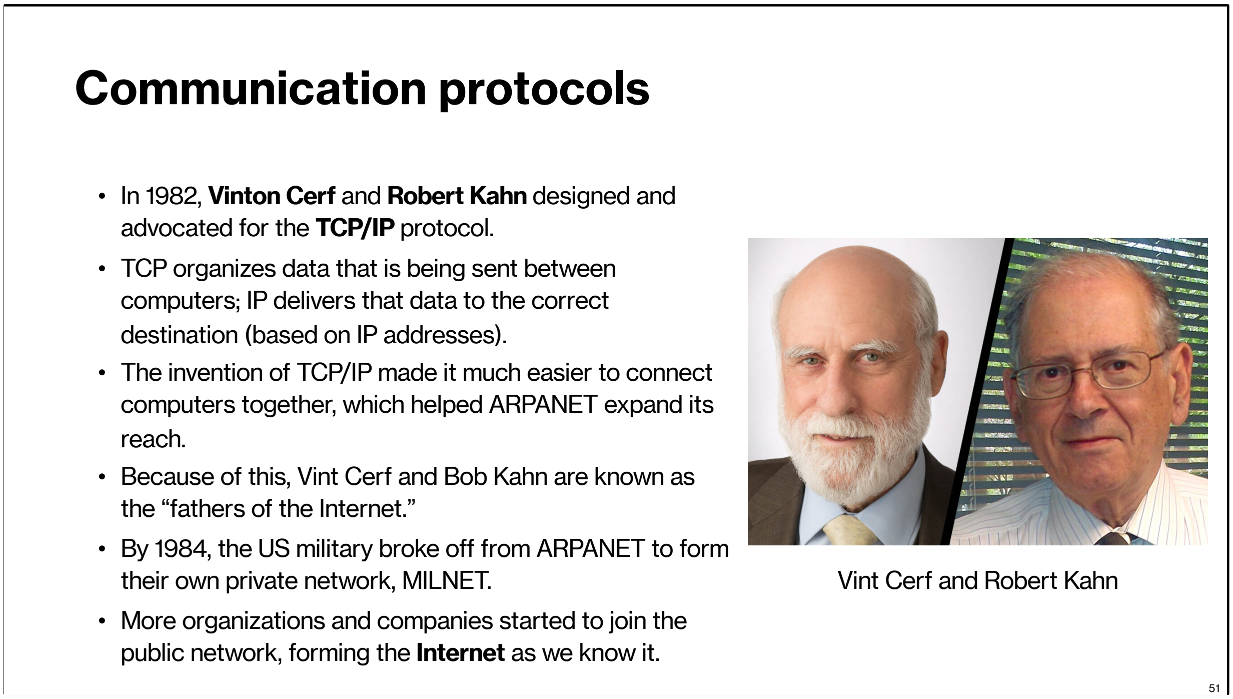
ARPANET

- In 1969, the US military wanted to create a decentralized communication system so that communications could not be knocked out entirely by a nuclear attack.
- DARPA (Defense Advanced Research Projects Agency) collaborated with several universities to build the **ARPANET**, the Advanced Research Projects Agency Network.
- First connection between UCLA and Stanford, but it grew quickly.
- The first email was sent over ARPANET in 1971!

ARPANET

50

The origins of the Internet can be traced back to ARPANET. In 1969, the US military wanted to create a decentralized communication system so that communications could not be knocked out entirely by a nuclear attack. DARPA (Defense Advanced Research Projects Agency, back then it was known as the Advanced Research Projects Agency) collaborated with several universities to build the **ARPANET**, the Advanced Research Projects Agency Network. The first connection at the end of 1969 was between UCLA and Stanford and by the end of the year four campuses had connected. Cool fact about ARPANET. The first email was sent over it in 1971 and a lot of the ARPANET traffic eventually ended up being emails.



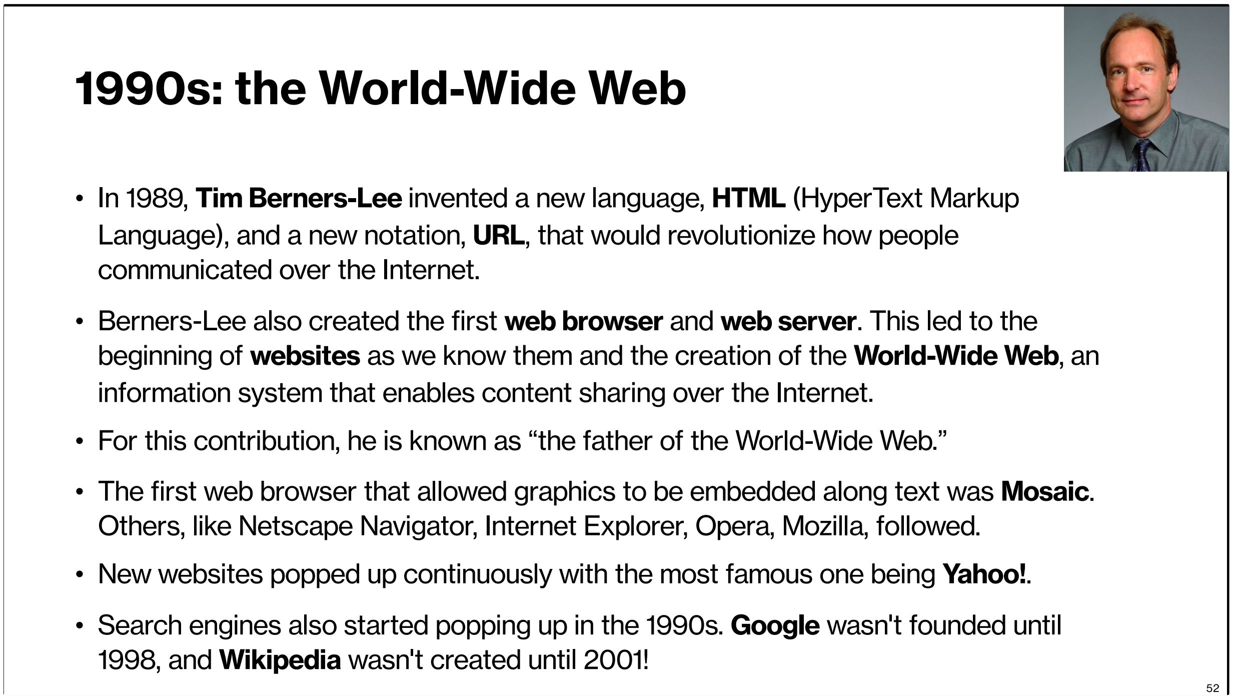
Communication protocols

- In 1982, **Vinton Cerf** and **Robert Kahn** designed and advocated for the **TCP/IP** protocol.
- TCP organizes data that is being sent between computers; IP delivers that data to the correct destination (based on IP addresses).
- The invention of TCP/IP made it much easier to connect computers together, which helped ARPANET expand its reach.
- Because of this, Vint Cerf and Bob Kahn are known as the “fathers of the Internet.”
- By 1984, the US military broke off from ARPANET to form their own private network, MILNET.
- More organizations and companies started to join the public network, forming the **Internet** as we know it.

Vint Cerf and Robert Kahn

51

As computers started getting connected with each other, there was a need for the establishment of communication protocols. In 1982, Vinton Cerf (1943-) and Robert Kahn (1938-) designed and advocated for the TCP/IP protocol.
TCP organizes data that is being sent between computers; IP delivers that data to the correct destination (based on IP addresses!). The invention of TCP/IP made it much easier to connect computers together, which helped ARPANET expand its reach.
Because of this, Vint Cerf and Bob Kahn are known as the fathers of the Internet.
By 1984, the US military broke off from ARPANET to form their own private network (MILNET). More organizations and companies started to join the public network, forming the Internet as we know it.

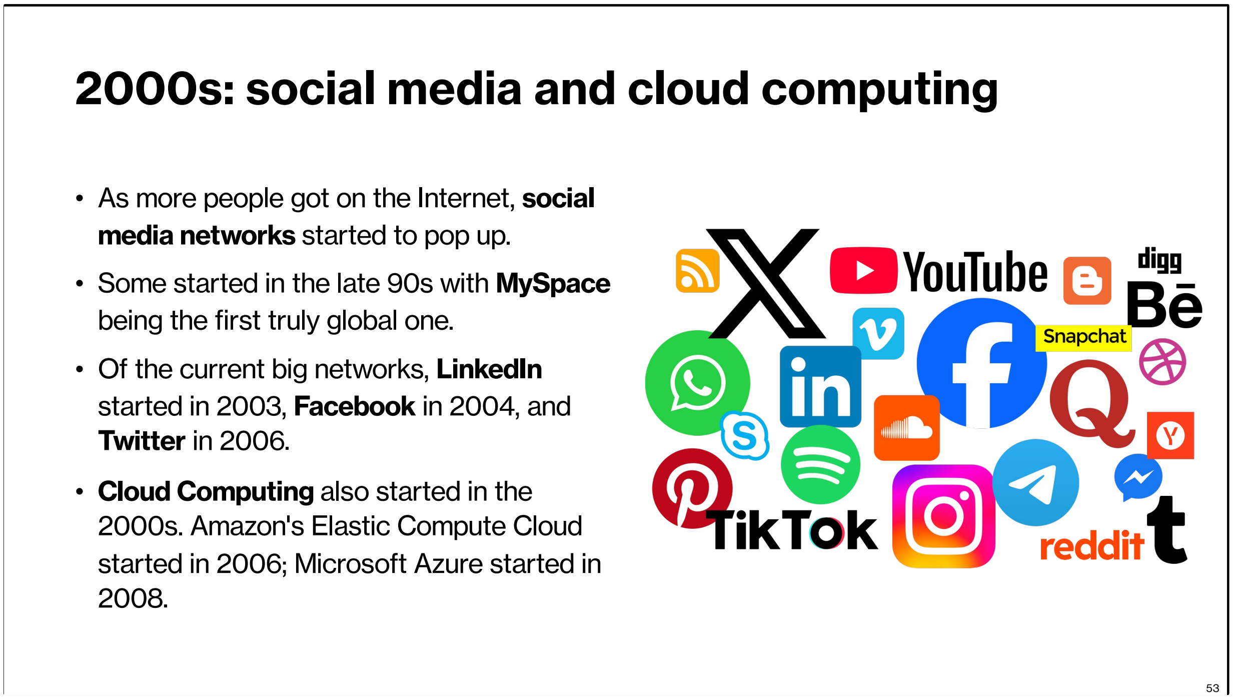


1990s: the World-Wide Web

- In 1989, **Tim Berners-Lee** invented a new language, **HTML** (HyperText Markup Language), and a new notation, **URL**, that would revolutionize how people communicated over the Internet.
- Berners-Lee also created the first **web browser** and **web server**. This led to the beginning of **websites** as we know them and the creation of the **World-Wide Web**, an information system that enables content sharing over the Internet.
- For this contribution, he is known as "the father of the World-Wide Web."
- The first web browser that allowed graphics to be embedded along text was **Mosaic**. Others, like Netscape Navigator, Internet Explorer, Opera, Mozilla, followed.
- New websites popped up continuously with the most famous one being **Yahoo!**.
- Search engines also started popping up in the 1990s. **Google** wasn't founded until 1998, and **Wikipedia** wasn't created until 2001!

52

In 1989, (now Sir) Tim Berners-Lee (1955-) invented a new language, HTML (HyperText Markup Language), and a new notation, URL, that would revolutionize how people communicated over the Internet. Berners-Lee who then worked at CERN also created the first web browser and web server. This led to the beginning of websites as we know them and the creation of the World-Wide Web, an information system that enables content sharing over the Internet. For this contribution, he is known as "the father of the World-Wide Web." The first browser that allowed graphics to be embedded along text was Mosaic. Others, like Netscape Navigator, Internet Explorer, Opera, Mozilla, followed, but fundamentally web browsers haven't changed much. New websites popped up continuously with the most famous one being Yahoo!. Search engines also started popping up in the 1990s. Google wasn't founded until 1998, and Wikipedia wasn't created until 2001!

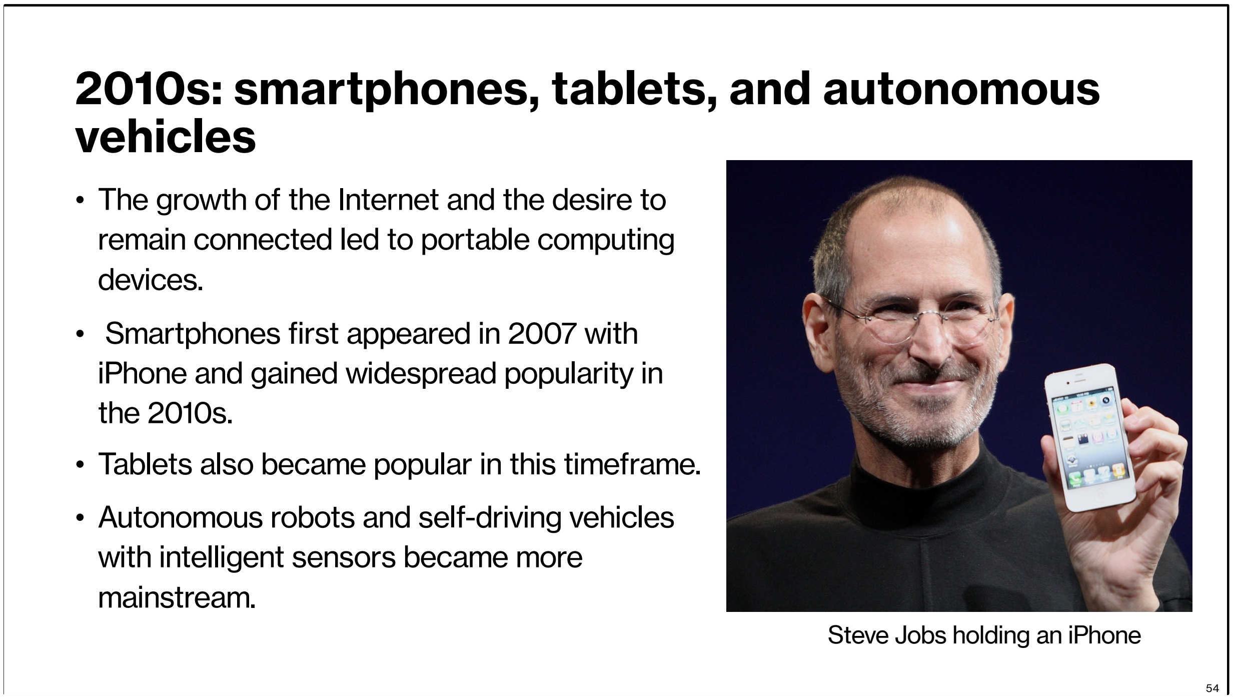


YouTube
digg
Bē
Snapchat
TikTok
reddit

2000s: social media and cloud computing

- As more people got on the Internet, **social media networks** started to pop up.
- Some started in the late 90s with **MySpace** being the first truly global one.
- Of the current big networks, **LinkedIn** started in 2003, **Facebook** in 2004, and **Twitter** in 2006.
- **Cloud Computing** also started in the 2000s. Amazon's Elastic Compute Cloud started in 2006; Microsoft Azure started in 2008.

The 2000s saw the introduction of social media as more and more people got on the Internet. MySpace was the one with a truly global outreach. Of the current big networks, LinkedIn started in 2003, Facebook in 2004, and Twitter in 2006. Cloud computing, where computing services are accomplished over the Internet, also took off in the 2000s.



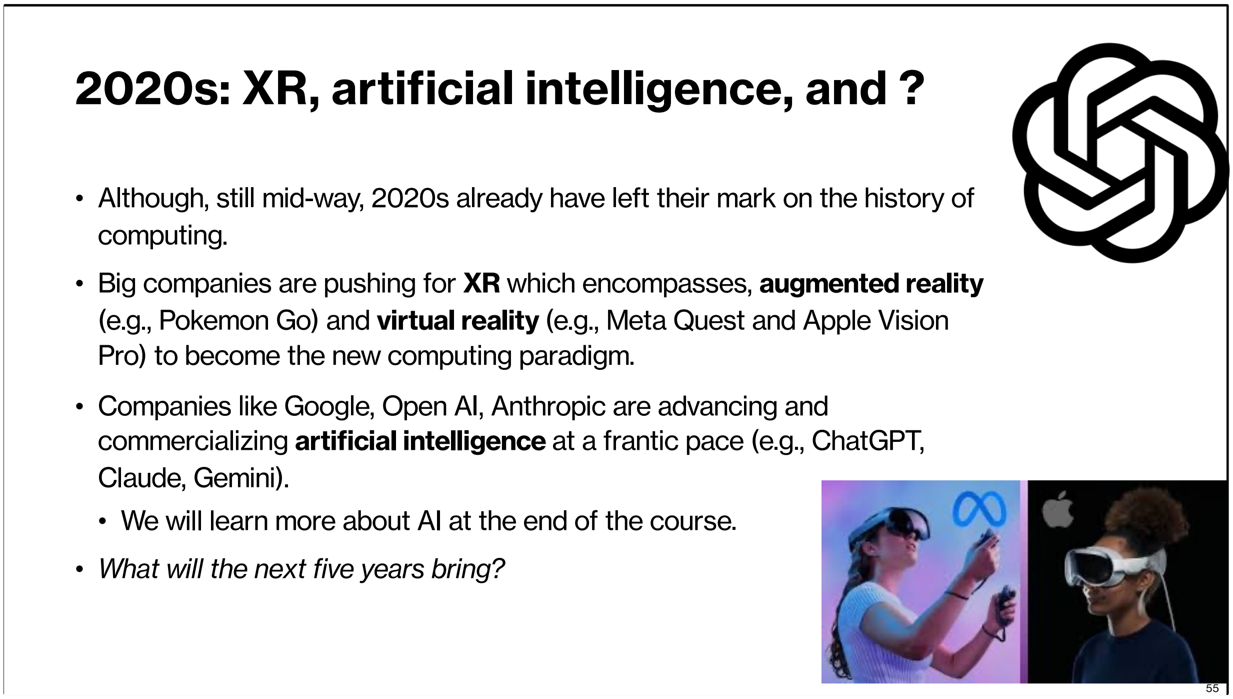
2010s: smartphones, tablets, and autonomous vehicles

- The growth of the Internet and the desire to remain connected led to portable computing devices.
- Smartphones first appeared in 2007 with iPhone and gained widespread popularity in the 2010s.
- Tablets also became popular in this timeframe.
- Autonomous robots and self-driving vehicles with intelligent sensors became more mainstream.

Steve Jobs holding an iPhone

54

The 2010s brought new devices and technologies into mass production. The growth of the Internet and the desire to remain connected led to portable computing devices. Smartphones first appeared in 2007 with the release of the iPhone and gained widespread popularity in the 2010s. Tablets also became popular in this timeframe. Autonomous robots and self-driving vehicles with intelligent sensors became more mainstream.

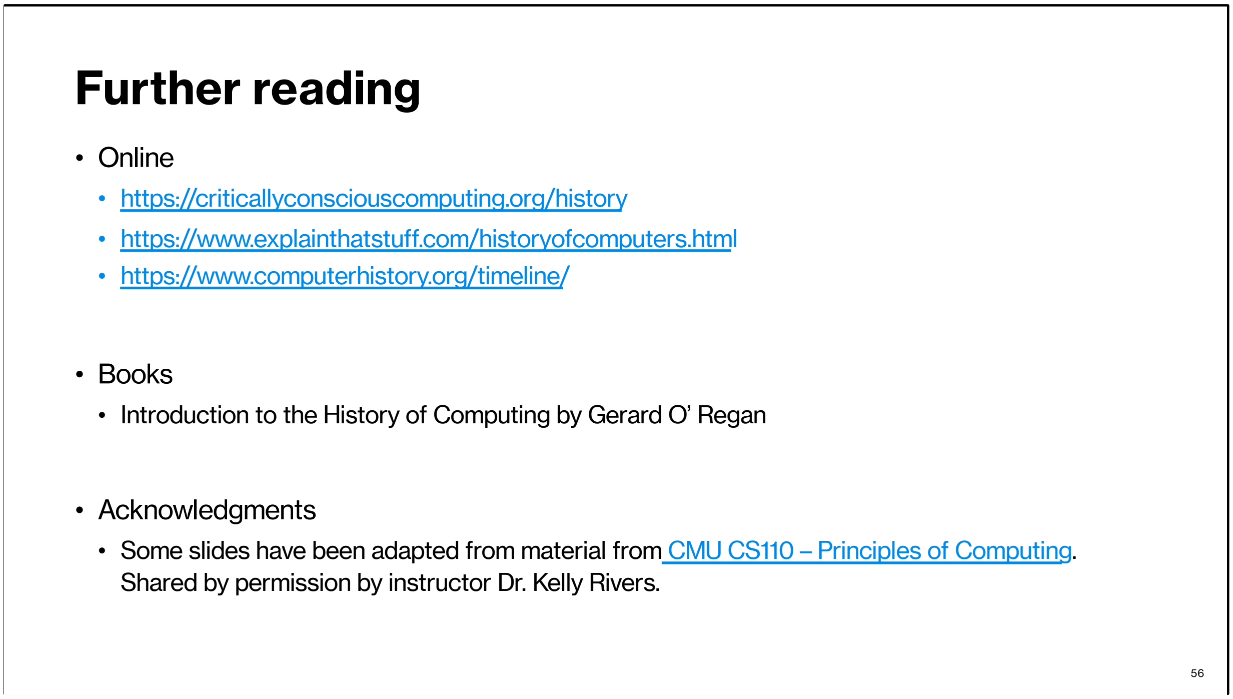


2020s: XR, artificial intelligence, and ?

- Although, still mid-way, 2020s already have left their mark on the history of computing.
- Big companies are pushing for **XR** which encompasses, **augmented reality** (e.g., Pokemon Go) and **virtual reality** (e.g., Meta Quest and Apple Vision Pro) to become the new computing paradigm.
- Companies like Google, Open AI, Anthropic are advancing and commercializing **artificial intelligence** at a frantic pace (e.g., ChatGPT, Claude, Gemini).
  - We will learn more about AI at the end of the course.
- *What will the next five years bring?*

55

Although, still mid-way, 2020s already have left their mark on the history of computing. Big companies are pushing for XR which encompasses, augmented reality (e.g., Pokemon Go) and virtual reality (e.g., Meta Quest and Apple Vision Pro) to become the new computing paradigm. Companies like Google, Open AI, Anthropic are advancing and commercializing artificial intelligence at a frantic pace (e.g., ChatGPT, Claude, Gemini).
What do you think the next five years will bring?



Further reading

- Online
  - https://criticallyconsciouscomputing.org/history
  - https://www.explainthatstuff.com/historyofcomputers.html
  - https://www.computerhistory.org/timeline/
- Books
  - Introduction to the History of Computing by Gerard O' Regan
- Acknowledgments
  - Some slides have been adapted from material from CMU CS110 – Principles of Computing. Shared by permission by instructor Dr. Kelly Rivers.

If you want to read more about the history of cs, here’s a few more resources.