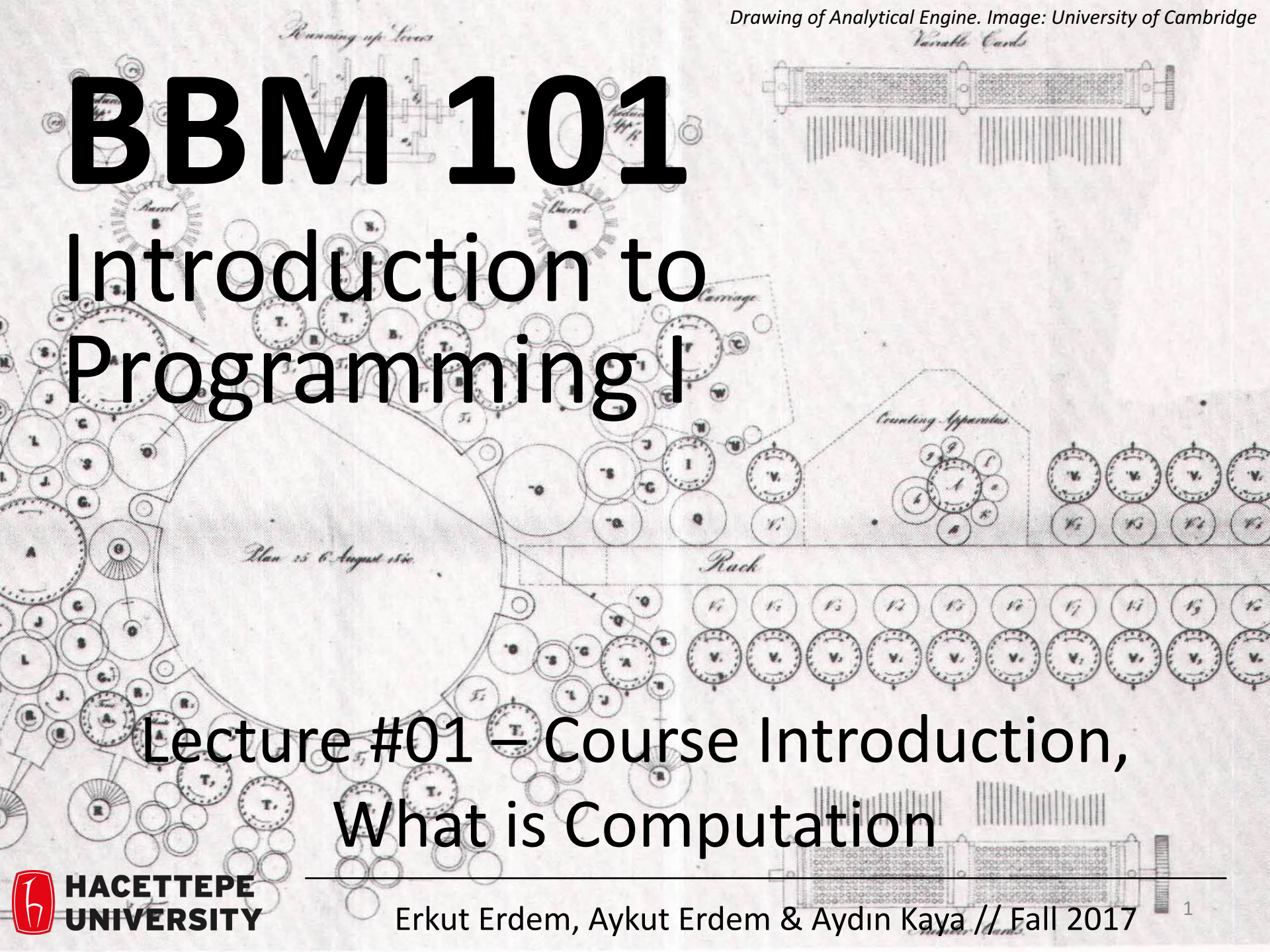


Drawing of Analytical Engine. Image: University of Cambridge

BBM 101

Introduction to Programming I

Lecture #01 – Course Introduction, What is Computation

HACETTEPE UNIVERSITY

Erkut Erdem, Aykut Erdem & Aydın Kaya // Fall 2017

Lecture Overview

- Course introduction

- What is computation

Disclaimer: Much of the material and slides for this lecture were borrowed from
— Ruth Anderson, Michael Ernst and Bill Howe's CSE 140 class

Course Information

Course Staff

- **Instructors:** Aykut Erdem, Erkut Erdem, Aydın Kaya



- **Teaching Assistants:**
  - Necva Bölücü
  - Selma Dilek
  - Selim Yılmaz
  - Cemil Zalluoğlu




Do not hesitate to ask TAs for help!

About BBM 101

- This course teaches core programming concepts with an emphasis on data manipulation tasks from science, engineering, and business

- **Goal** by the end of the semester: Given a data source and a problem description, you can independently write a complete, useful program to solve the problem

- **BBM103 Introduction to Programming Laboratory I**
  - Students will gain skills to apply the concepts to real world problems

Learning Objectives

- Computational problem-solving
  - Writing a program will become your “go-to” solution for data analysis tasks.

- Basic Python and C proficiency
  - Including experience with relevant libraries for data manipulation, scientific computing, and visualization.

What This Course is not

- A “skills course” in Python/C
  - …though you’ll become proficient in the basics of the Python/C programming language
  - …and you will gain experience with some important Python/C libraries

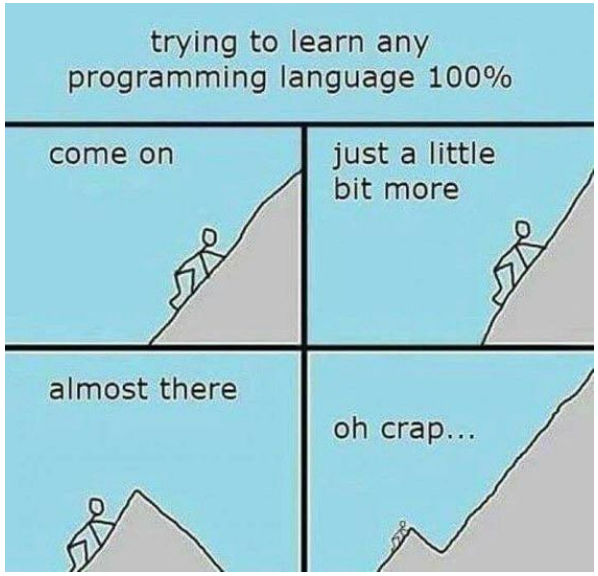


Image credit: Google+ user nixCraft

- A “project” course
  - the assignments are “real,” but are intended to teach specific programming concepts

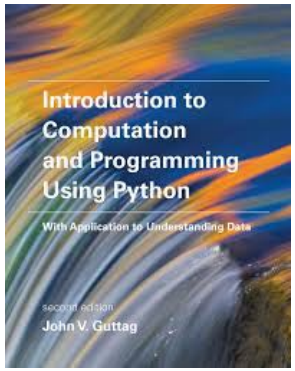
- A “software engineering” course
  - Programming is the starting point of computer science and software engineering

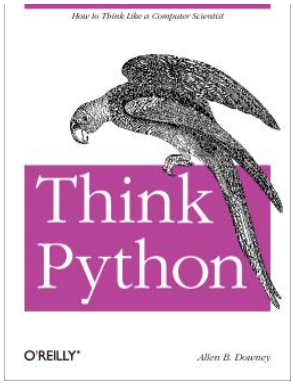
Communication

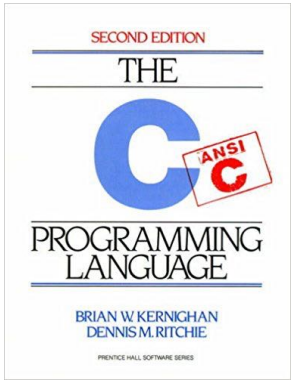
- Website: http://web.cs.hacettepe.edu.tr/~bbm101/
- See the website for all administrative details
- Read the handouts and required texts, *before* the lecture
- Take notes!
- Follow the course in Piazza https://piazza.com/hacettepe.edu.tr/fall2017/bbm101

Text Books

- The Python Tutorial, available from the Python website.
  - This is good for explaining the nuts and bolts of how Python works.
- Introduction to Computation and Programming Using Python, Second Edition, John V. Guttag, MIT Press, August 2016
- Think Python, 2nd edition
  - Freely available online in HTML and PDF.
  - Also available for purchase as a printed book, but don't buy the first edition.
  - This book introduces more conceptual material, motivating computational thinking.
- There is an interactive version of "How to Think Like a Computer Scientist" (the first edition of "Think Python"), which lets you type and run Python code directly while reading the book.
- The C Programming Language, 2nd Edition, Brian Kernighan and Dennis Ritchie, Prentice Hall, 1988







Grading Policy

- Grading for BBM 101 will be based on
  - two midterm exam (25+30=55%)
  - a final exam (40%)
  - class participation (5%)

- In BBM 103, the grading will be based on
  - five assignments (75%)
  - a set of quizzes (25%) (Your lowest 2 quiz grades will be dropped.)

Academic Integrity

- Honest work is required of a scientist or engineer.
- Collaboration policy on the course web. **Read it!**
  - Discussion is permitted.
  - **Carrying materials from discussion is not permitted.**
  - Everything you turn in must be your own work.
    - Cite your sources, explain any unconventional action.
  - **You may not view others' work.**
  - If you have a question, ask.
- We trust you completely.
- But we have no sympathy for trust violations – nor should you!

How to Succeed

- No prerequisites
- Non-predictors for success:
  - Past programming experience
  - Enthusiasm for games or computers
- Programming and data analysis are challenging
- Every one of you can succeed
  - There is no such thing as a **"born programmer"**
  - Work hard
  - Follow directions
  - Be methodical
  - *Think* before you act
  - Try on your own, then ask for help
  - Start early

Python

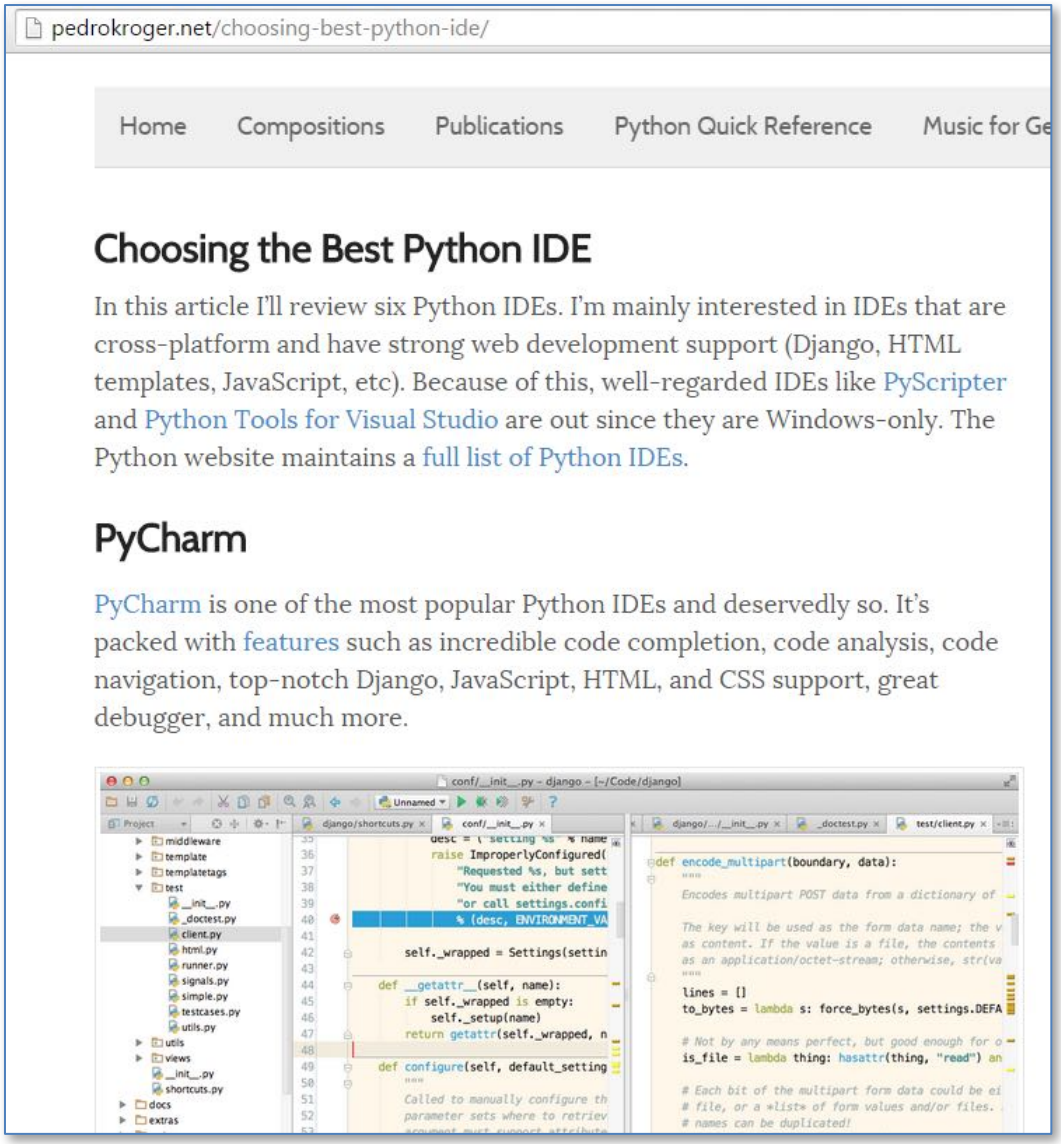
Python Version

- Whatever IDE you choose to work with, always stick to **Python version 3.5.2**

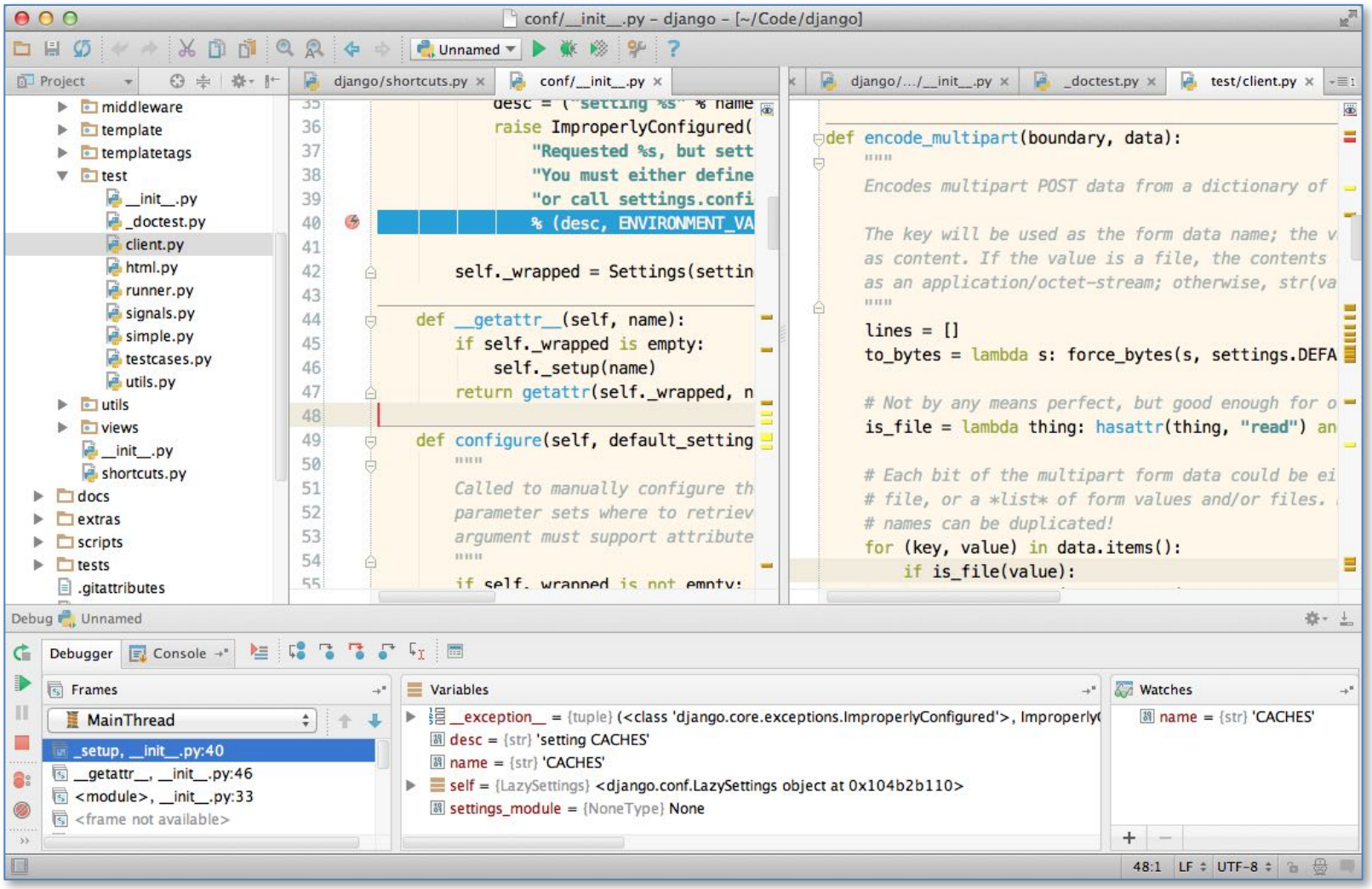
- **Always** use this version to code your assignments.

Integrated Development Environment (IDE)

- There are many!



Our Recommendation: PyCharm



Computer Programming

"It's a great time to be a data geek."

-- Roger Barga, Microsoft Research

"The greatest minds of my generation are trying to figure out how to make people click on ads"

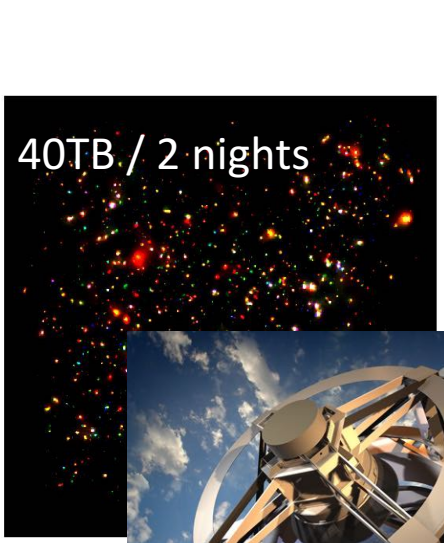
-- Jeff Hammerbacher, co-founder, Cloudera

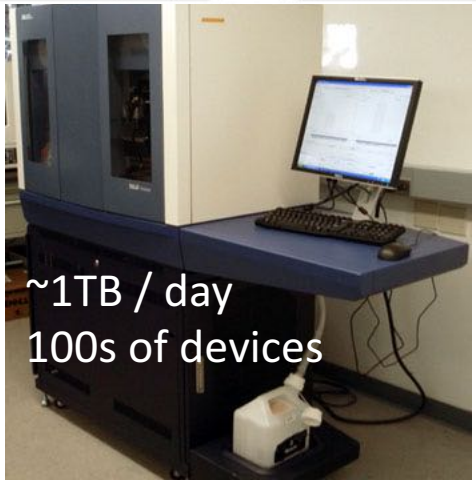
All of Science is Reducing to Computational Data Manipulation

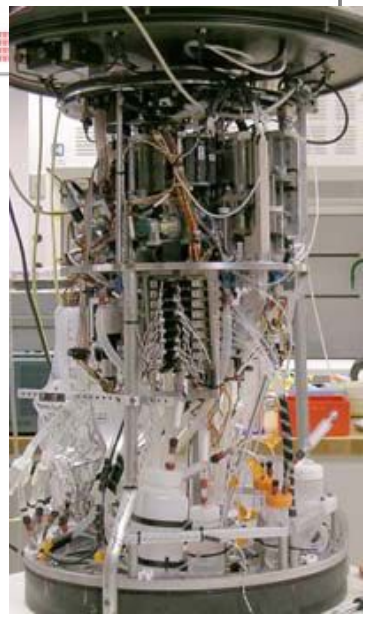
Old model: "Query the world" (Data acquisition coupled to a specific hypothesis)

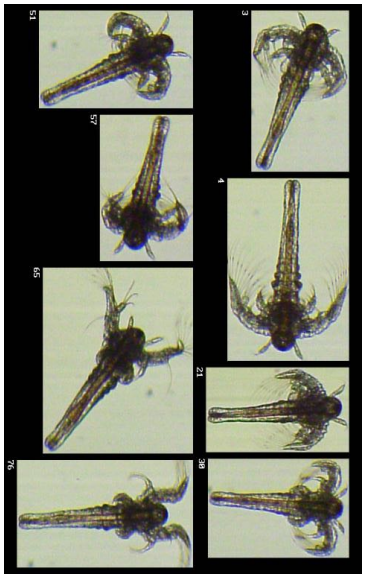
New model: "Download the world" (Data acquisition supports many hypotheses)

- Astronomy: High-resolution, high-frequency sky surveys (SDSS, LSST, PanSTARRS)
- Biology: lab automation, high-throughput sequencing,
- Oceanography: high-resolution models, cheap sensors, satellites









Example: Assessing Treatment Efficacy

	A	B	C	D	E	F	G	H	I	J
1	fu_2wk	fu_4wk	fu_8wk	fu_12wk	fu_16wk	fu_20wk	fu_24wk	total4type_fu	clinic_zip	pt_zip
2	1	3	4	7	9	9	9	12	98405	98405
3	2	4	6	7	8	8	8	8	98405	98403
4	0	0	0	0	0	0	0	0	98405	98445
5	3	4	5	5	5	5	5	5	98405	98332
6	0	0	0	0	0	0	0	0	98405	98405
7	2	2	2	2	2	2	2	2	98402	98402
8	1	2	5	6	8	10	10	14	98405	98418
9	1	1	2	2	2	2	2	2	98499	98406
10	0								98405	98404
11	0								98405	98402
12	1								98405	98405
13	1								98404	98404
14	2								98499	98498
15	0	0	0	0	0	0	0	0	98499	98445
16	1	2	4	5	7	7	7	7	98499	98405
17	1	1	1	2	2	2	2	2	98499	98498

number of follow ups within 16 weeks after treatment enrollment.

Zip code of clinic

Zip code of patient

Question: Does the distance between the patient's home and clinic influence the number of follow ups, and therefore treatment efficacy?

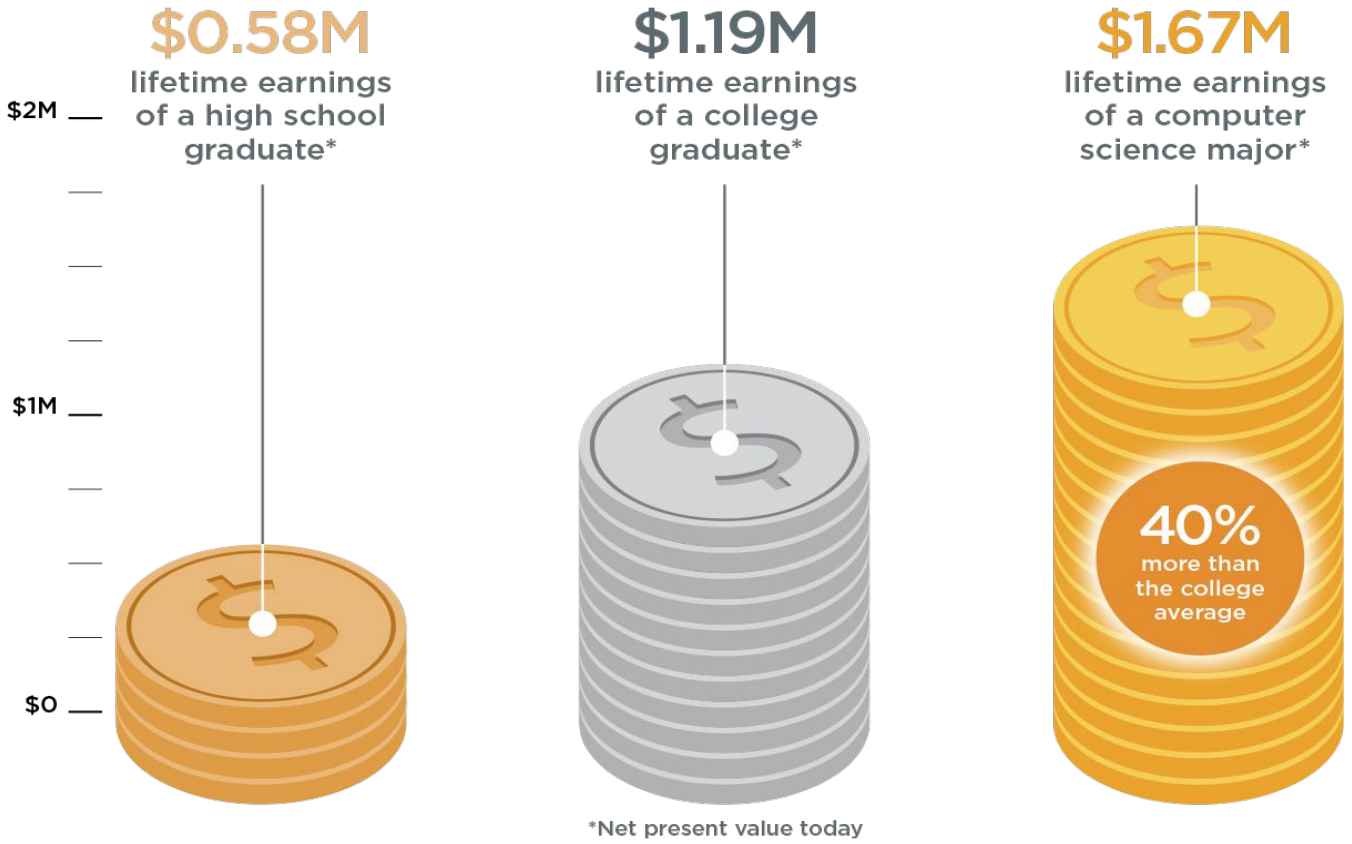
Python Program to Assess Treatment Efficacy

```
# This program reads an Excel spreadsheet whose penultimate
# and antepenultimate columns are zip codes.
# It adds a new last column for the distance between those zip
# codes, and outputs in CSV (comma-separated values) format.
# Call the program with two numeric values:  the first and last
# row to include.
# The output contains the column headers and those rows.

# Libraries to use
import random
import sys
import xlrd          # library for working with Excel
spreadsheets
import time
from gdapi import GoogleDirections

# No key needed if few queries
gd = GoogleDirections('dummy-Google-key')

wb = xlrd.open_workbook('mhip_zip_eScience_121611a.xls')
sheet = wb.sheet_by_index(0)

# User input:  first row to process, first row not to process
first_row = max(int(sys.argv[1]), 2)
row_limit = min(int(sys.argv[2]+1), sheet.nrows)

def comma_separated(lst):
  return ",".join([str(s) for s in lst])

headers = sheet.row_values(0) + ["distance"]
print comma_separated(headers)

for rownum in range(first_row,row_limit):
    row = sheet.row_values(rownum)
    (zip1, zip2) = row[-3:-1]
    if zip1 and zip2:
        # Clean the data
        zip1 = str(int(zip1))
        zip2 = str(int(zip2))
        row[-3:-1] = [zip1, zip2]
        # Compute the distance via Google Maps
        try:
            distance = gd.query(zip1,zip2).distance
        except:
            print >> sys.stderr, "Error computing distance:",
zip1, zip2
            distance = ""
      # Print the row with the distance
      print comma_separated(row + [distance])
      # Avoid too many Google queries in rapid succession
      time.sleep(random.random()+0.5)
```

23 lines of executable code!

Some statistics (from U.S.)

The value of a computer science education

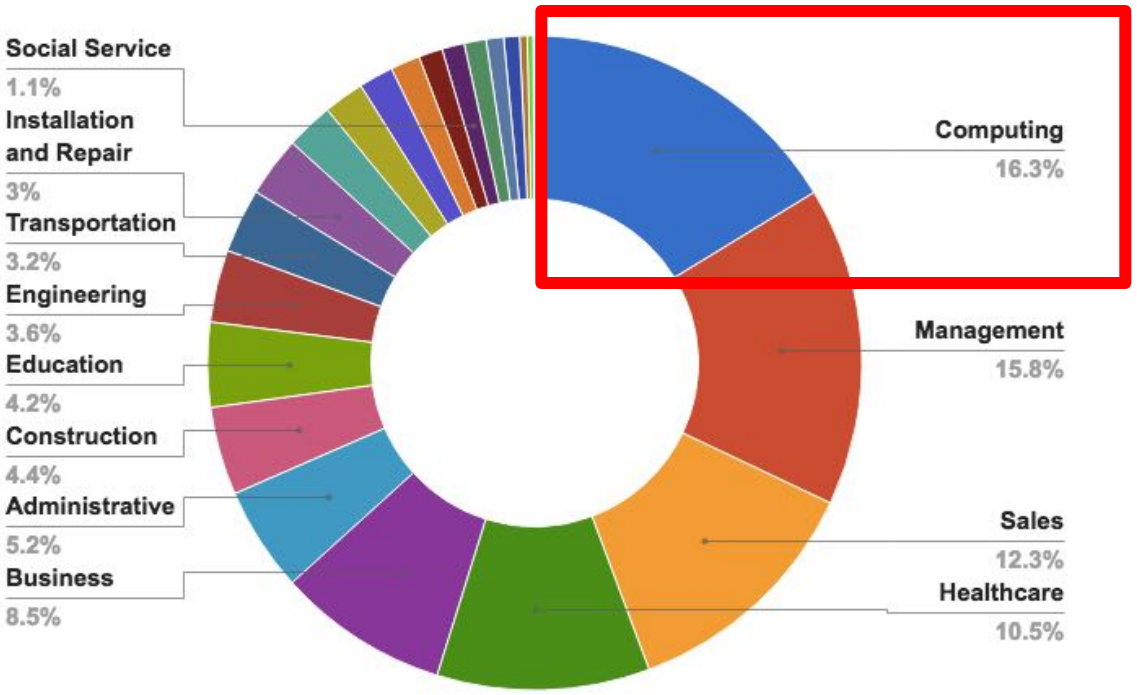


Source: Brookings

Slide credit: code.org

Some statistics (from U.S.)

Computing jobs are the #1 source of new wages in the United States

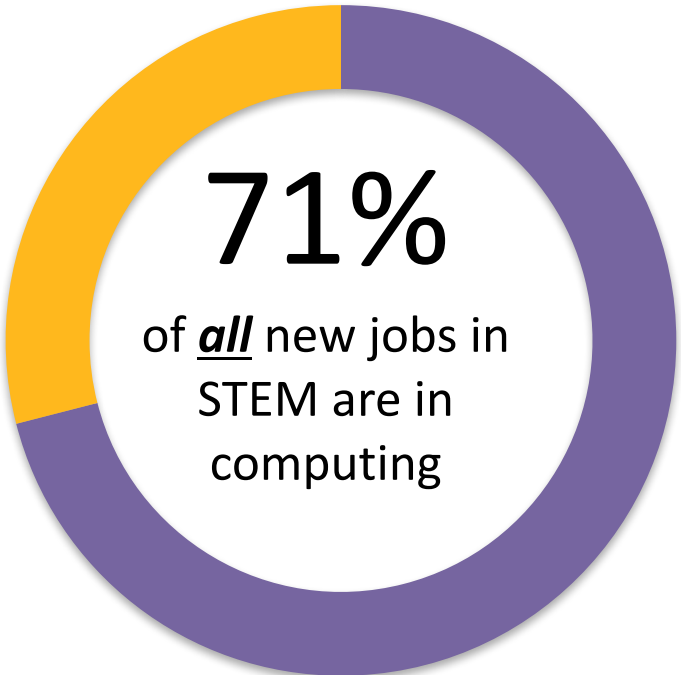


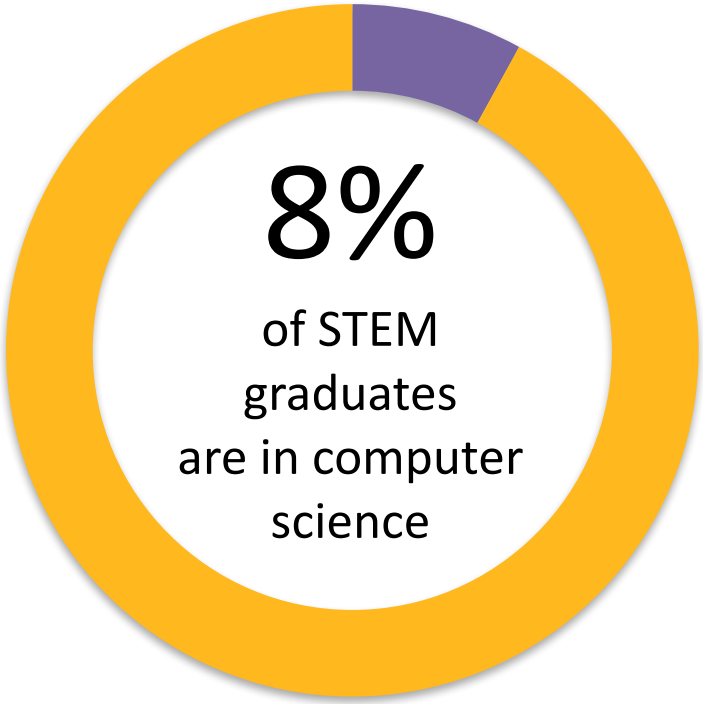
500,000 current openings: These jobs are in ***every*** industry and ***every*** state, and they're projected to grow at twice the rate of all other jobs.

Slide credit: code.org

Some statistics (from U.S.)

The STEM* problem **is in computer science:**





Sources: Bureau of Labor Statistics, National Center for Education Statistics

Slide credit: code.org

*STEM = Science, Technology, Engineering, and Math

What is Computation?

Some may think:

Computer science is just about learning technology

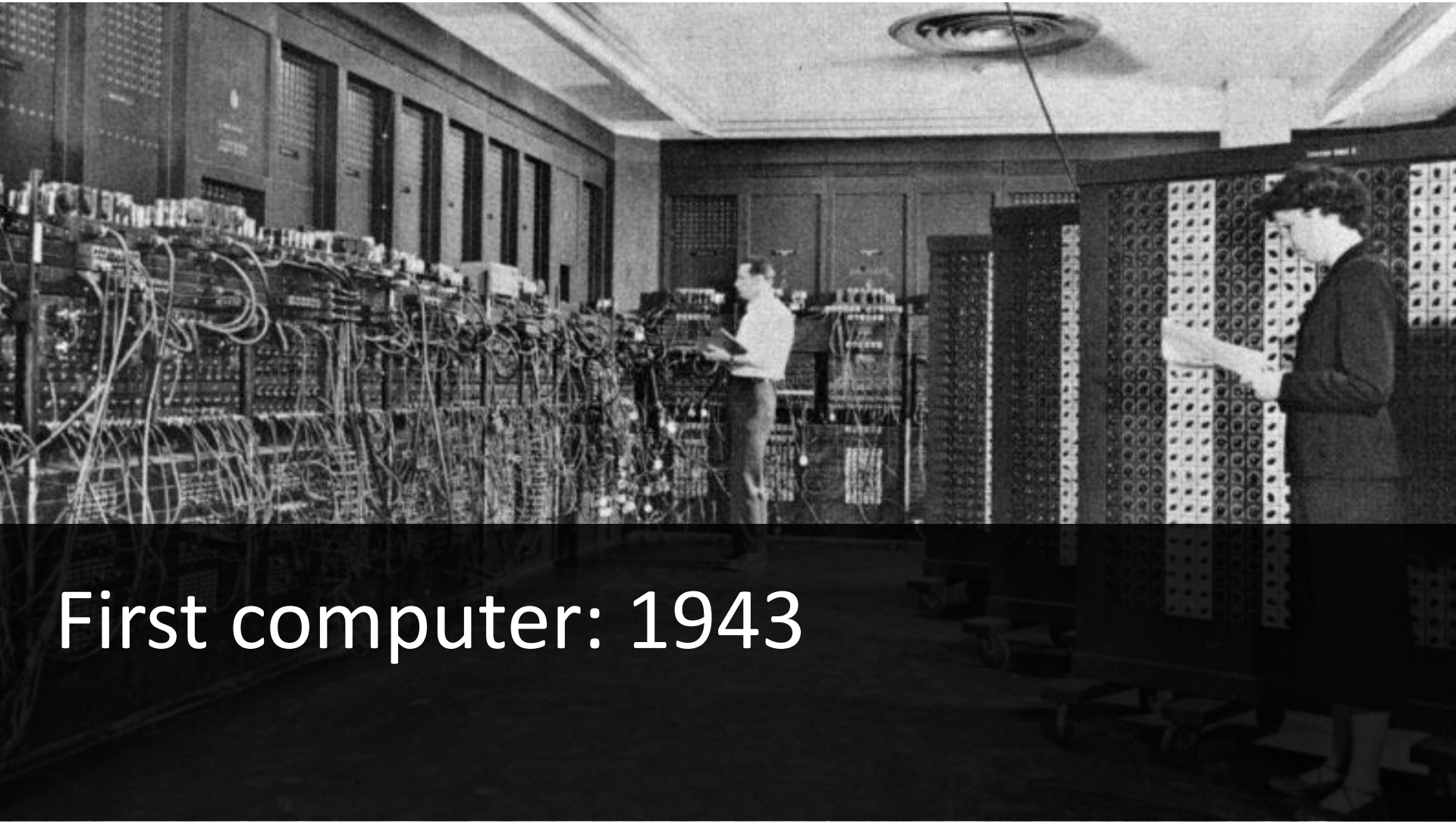
Slide credit: code.org

Some may think:

~~Computer science is just about learning technology~~

Computer science is about logic, problem solving, and creativity

Slide credit: code.org



First computer: 1943

Slide credit: code.org

First computer: 1943

First computer program: 1843

Slide credit: code.org

What is Knowledge?

- **Declarative knowledge**
  - Axioms (definitions)
  - Statements of fact

"y is the square root of x if and only if y*y = x"

does not help to find the square root!

What is Knowledge? (cont'd.)

- **Declarative knowledge**
  - Axioms (definitions)
  - Statements of fact

"y is the square root of x if and only if y*y = x"

does not help to find the square root!

- **Imperative knowledge**
  - How to do something
  - A sequence of specific instructions (what computation is about)

Babylonian method

Get x as an input

1. Begin with an arbitrary positive number y_0 *(an initial guess)*
2. If $y_n^2 \approx x$, stop *(found the solution - y_n)*
   Else let $y_{n+1} = (y_n + x/y_n)/2$ *(use the arithmetic mean to approximate the geometric mean)*
3. Repeat step (2)

What is Knowledge? (cont'd.)

- **Another example** – Estimating greatest common divisor (gcd)

Declarative definition

"d is the gcd of a and b if and only if d is the largest possible integer satisfying a = d*x and b = d*y with x and y being two positive integers"

Imperative definition: Euclid's formula

Get 2 positive integers a and b, a>=b as input

1. Divide a by b, call the remainder R
2. If R = 0, stop
   Else let a = b and b = R
3. Repeat step 2

(found the solution - b)

Use Euclid's formula to compute gcd(48,18).

What is a Computer?

- A device that executes a sequence of computations and instructions.
- Modern computers are electronic and digital.
- Does pencil and paper count as a computer?

Slide credit: B. Huang

Programs

- These sequences of instructions and computations is called a **program**.

- We will be designing programs in this course.

- These programs will be based on **algorithms**.
  - **Algorithm** - a step-by-step problem-solving procedure.

Slide credit: B. Huang

Where did the Term ‘Computer’ Originate?

- The definition from The Oxford Dictionary:

  ***“Computer (noun)**. A person who makes calculations, especially with a calculating machine.”*

Courtesy of the Library of Congress

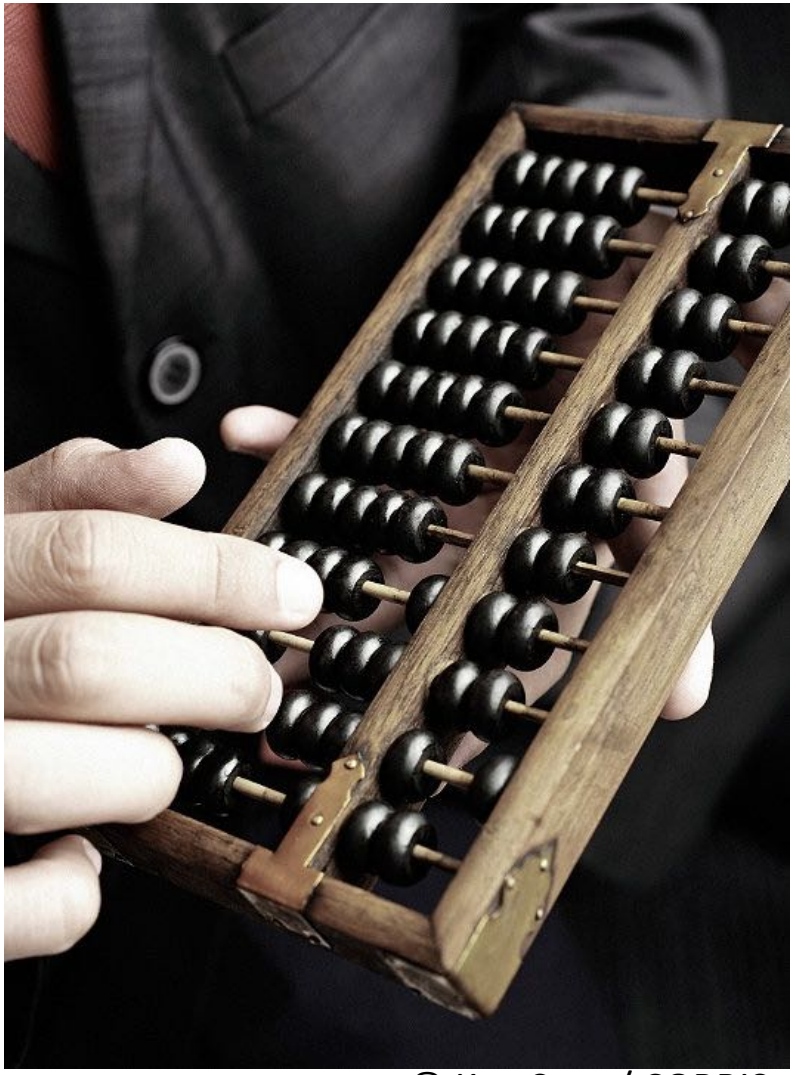
Fixed Program Computers

- Developed to solve a specific problem (set).

- Very old roots, old perspectives, …
  - Abacus
  - Antikythera Mechanism
  - Pascaline
  - Leibniz Wheel
  - Jacquard's Loom
  - Babbage Difference Engine
  - The Hollerith Electric Tabulating System
  - Atanasoff-Berry Computer (ABC)
  - Turing Bombe
  - etc.

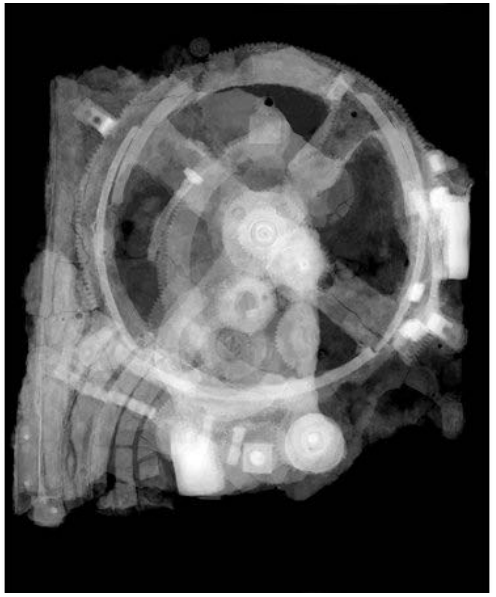
Abacus (500 BC)

- First pocket calculator
- Still used by businessmen in Asia.



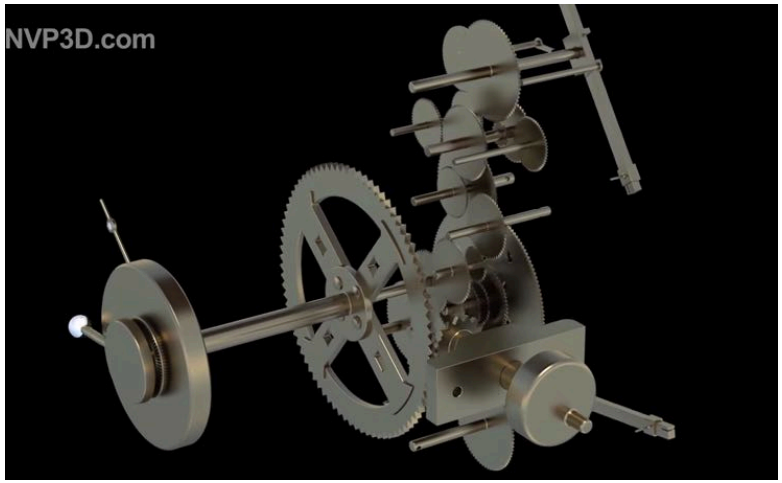
Antikythera Mechanism (100 BC)

- First analog computer
- An ancient mechanical computer designed to calculate astronomical positions



© Antikythera Mechanism Research Project

© Rien van de Weygaert



Pascaline (1642)

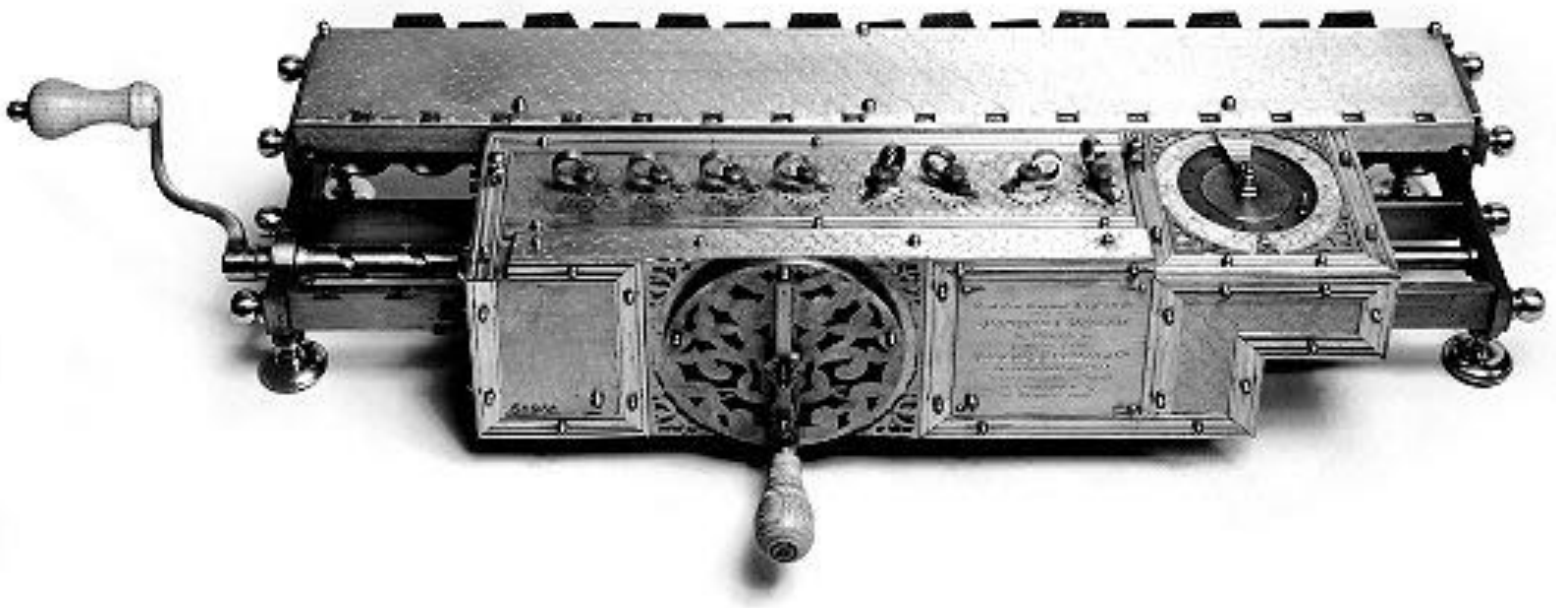
- Blaise Pascal, 1642
- A mechanical calculator for performing two arithmetic operations: addition and subtraction

© Britannica

© Mark Richards

Leibniz Wheel (1694)

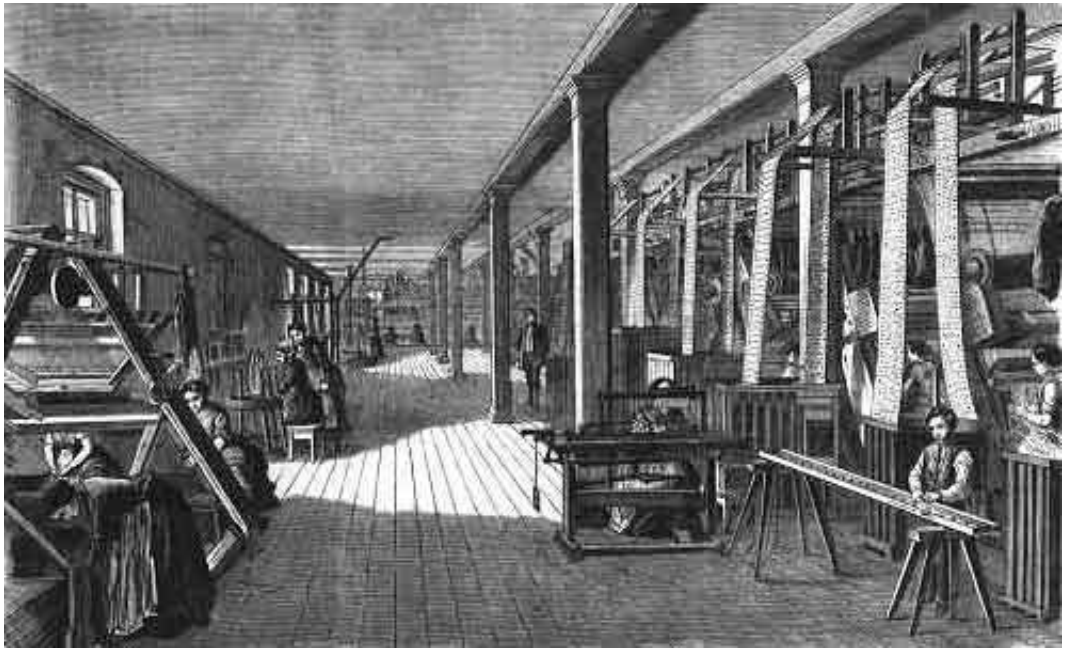
- Gottfried Wilhelm von Leibniz, 1694
- A mechanical calculator for performing all four arithmetic operations: addition, subtraction, multiplication and division



Courtesy of the Deutsches Museum, München

Jacquard's Loom (1801)

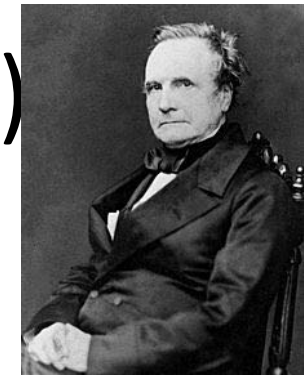
- Developed in 1801 by Joseph-Marie Jacquard.
- The loom was controlled by a loop of punched cards.
- Holes in the punched cards determined how the knitting proceeded, yielding very complex weaves at a much faster rate



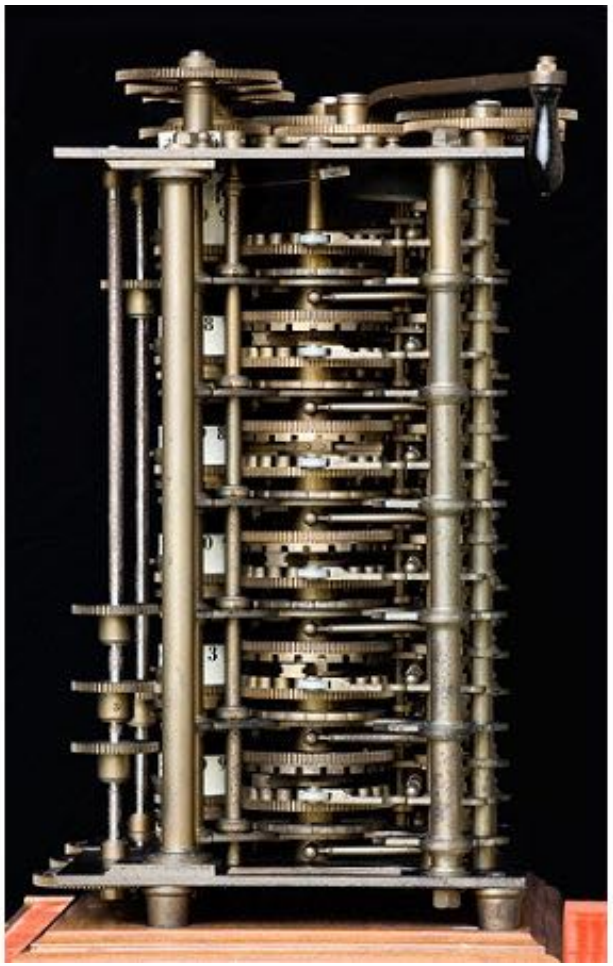
A Jacquard Loom workshop - Germany, 1858.

Adopted from: T. Cortina

Babbage Difference Engine (1832)



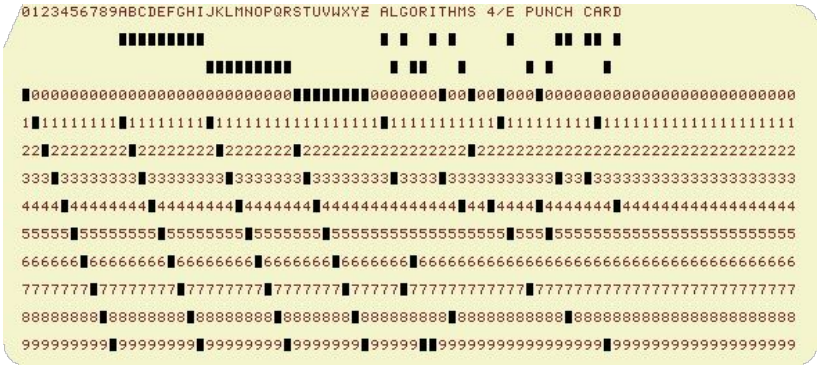
- Charles Babbage, 1832
- A mechanical calculator designed to tabulate polynomial functions (can be used for solving polynomial equations, curve fitting, etc.)
- A working difference engine was built in 1991 to celebrate the 200th anniversary of Babbage's birth (London Science Museum).
- It could hold 8 numbers of 31 decimal digits each and could thus tabulate 7th degree polynomials to that precision.



The Hollerith Electric Tabulating System

- **1880 Census.** Took 1,500 people 7 years to manually process data.
- **Herman Hollerith.** Developed counting and sorting machine to automate.
  - Use punch cards to record data (e.g., gender, age).
  - Machine sorts one column at a time (into one of 12 bins).
  - Typical question: how many women of age 20 to 30?

Hollerith tabulating machine and sorter



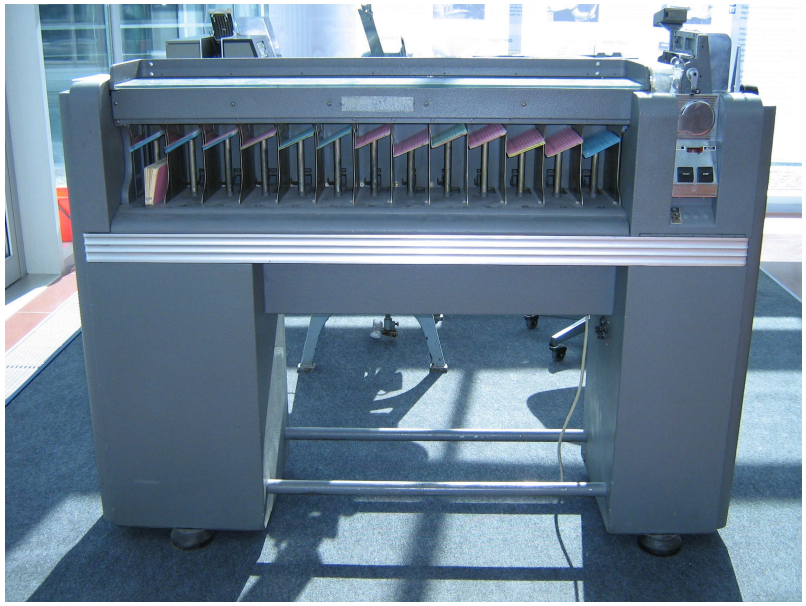
punch card (12 holes per column)

- 1890 Census. Finished months early and under budget!

Modern Punch Cards

- Punch cards. [1900s to 1950s]
  - Also useful for accounting, inventory, and business processes.
  - Primary medium for data entry, storage, and processing.

- Hollerith's company later merged with 3 others to form Computing Tabulating Recording Corporation (CTRC); the company was renamed in 1924.

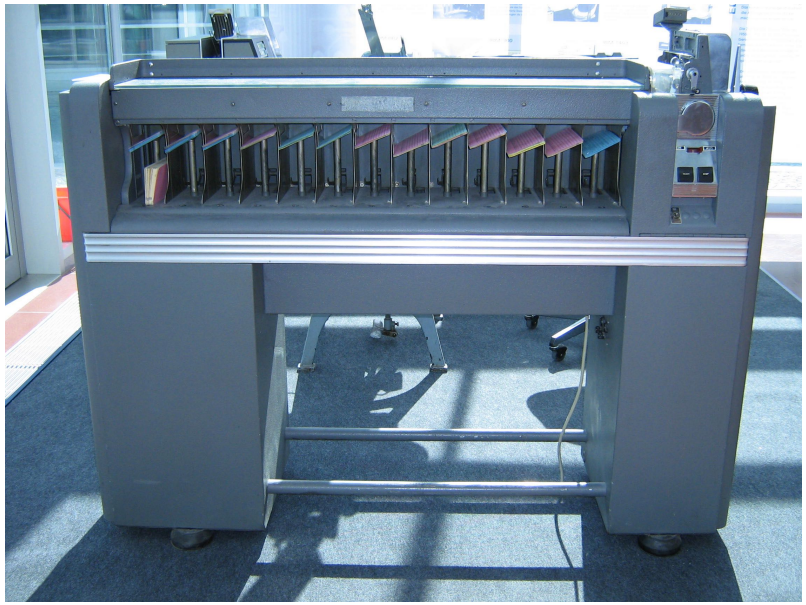


Adopted from: Sedgewick and Wayne

Modern Punch Cards

- Punch cards. [1900s to 1950s]
  - Also useful for accounting, inventory, and business processes.
  - Primary medium for data entry, storage, and processing.

- Hollerith's company later merged with 3 others to form Computing Tabulating Recording Corporation (CTRC); the company was renamed in 1924.

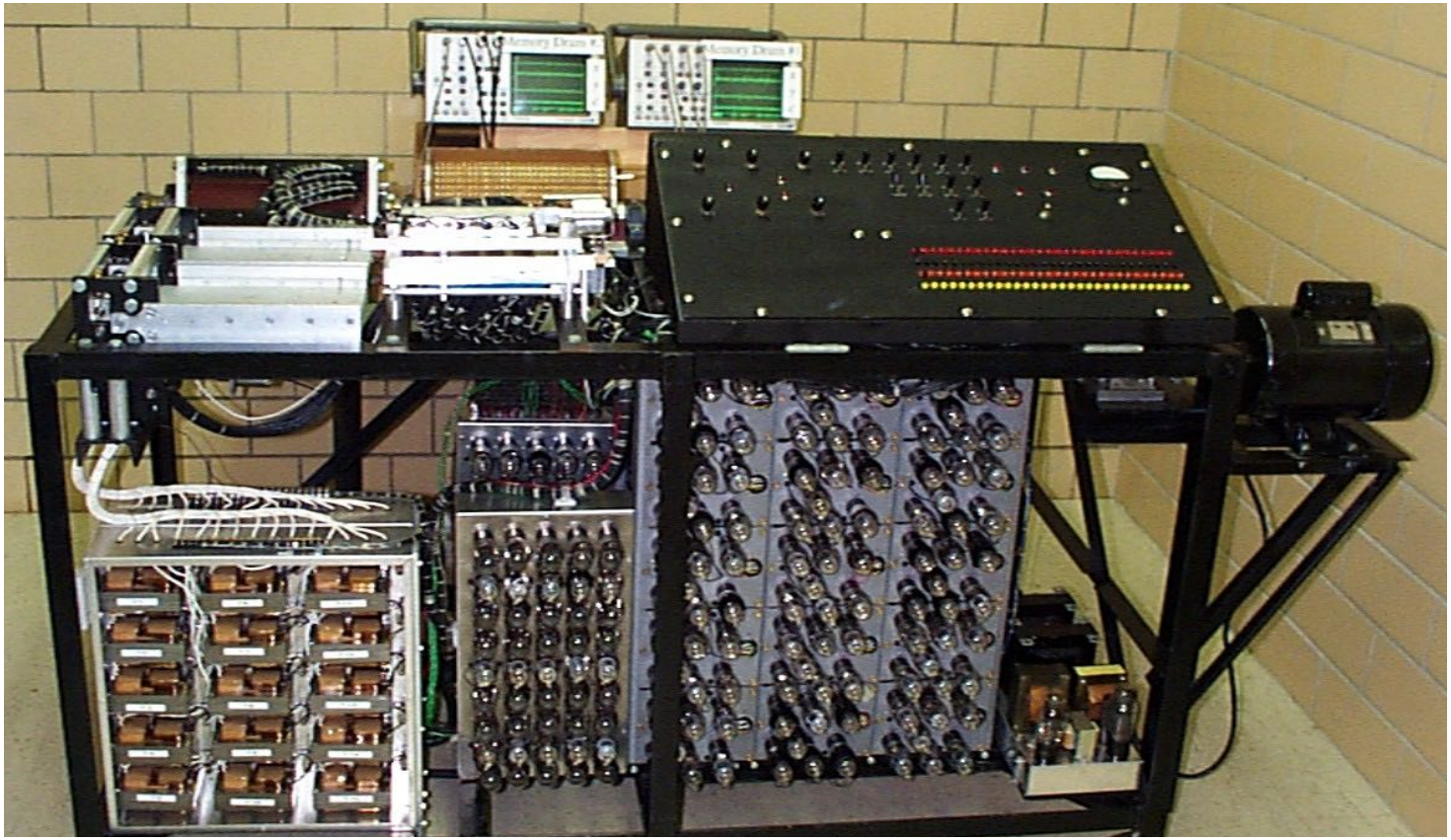


IBM 80 Series Card Sorter, 1949
(650 cards per minute)

Adopted from: Sedgewick and Wayne

Atanasoff-Berry Computer (ABC) (1939)

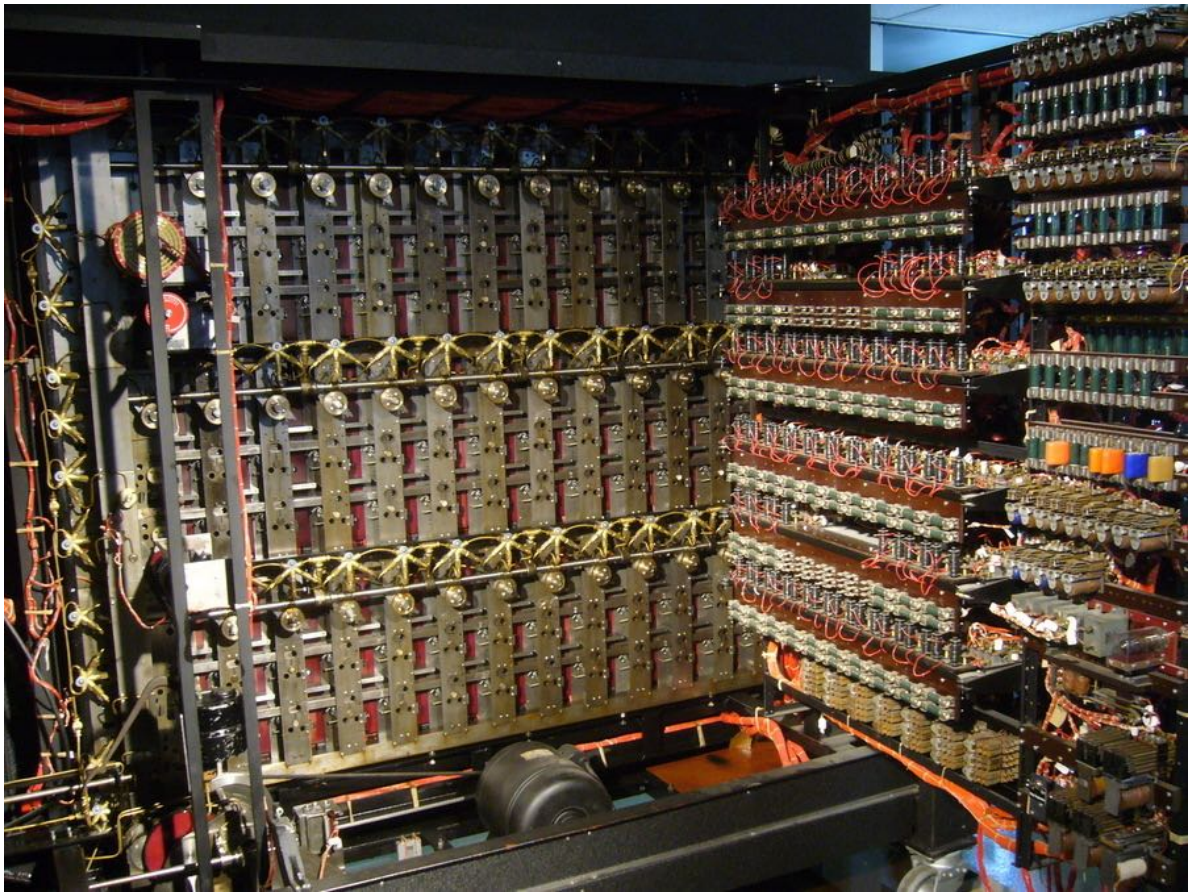
- John Vincent Atanasoff and Clifford Berry, 1939-1942
- One of the first electronic digital computing devices
- Designed to solve a system of linear equations



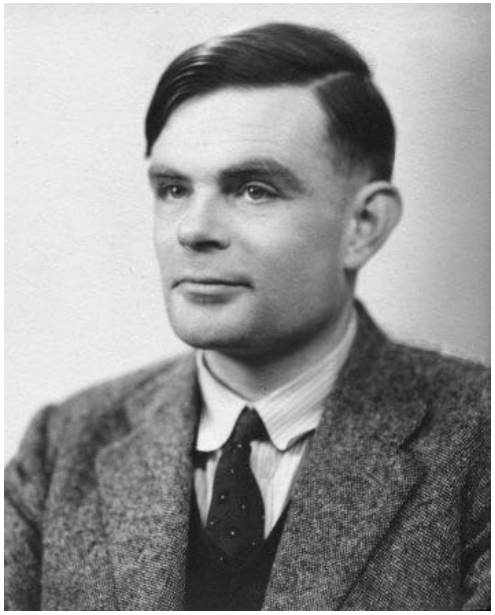
Turing Bombe (1941)

- Alan Turing, 1939
- Developed to crack German Enigma codes during WW II.

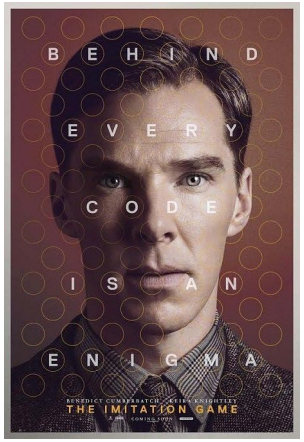
Enigma machine in use



Alan Turing



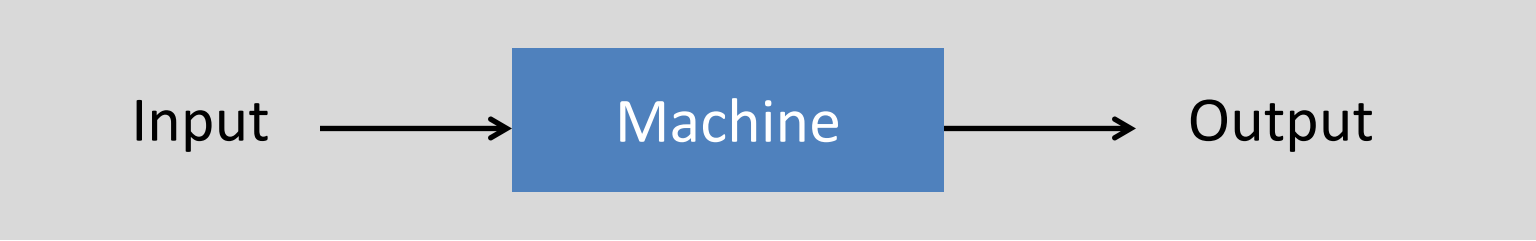
- 1912-1954
- Considered the “father” of modern computer science.
- Presented formalisms for the notions of computation and computability in the 1930’s.
- Worked at Bletchley Park in Great Britain during WWII to develop Collossus to help break the German Enigma Code.
- Developed the notion in 1950 of a test for machine intelligence now called the Turing Test.
- The Turing Award, the highest award in computing, is named in honor of Alan Turing.



Slide credit: G. Kesden

Stored Program Computers

- Problem solving



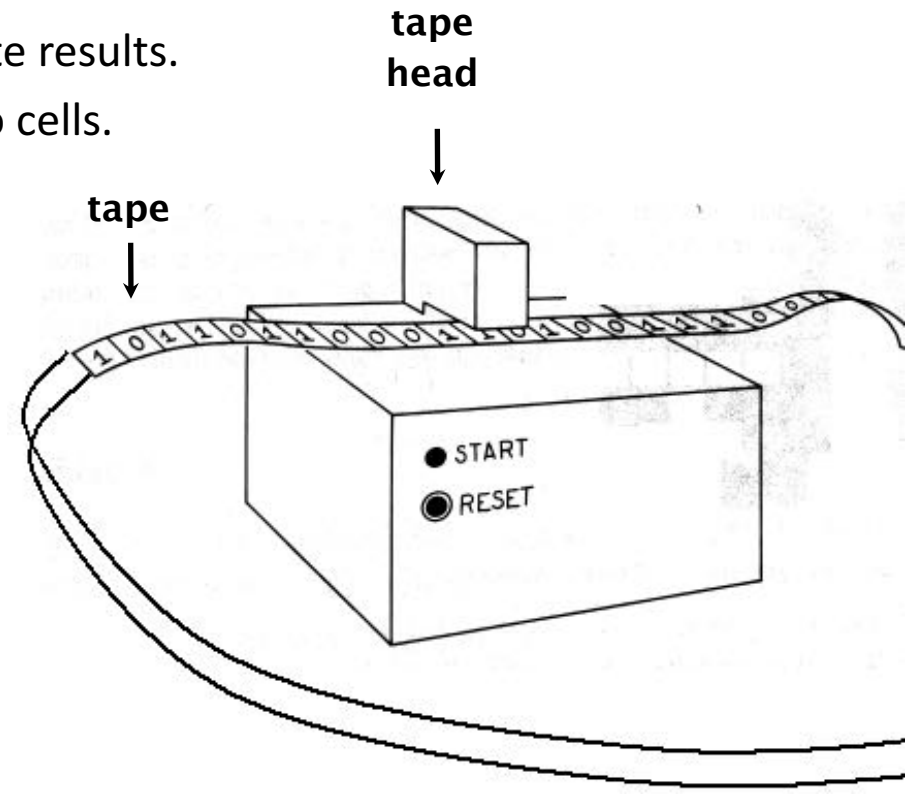
- What if input is a machine (description) itself?

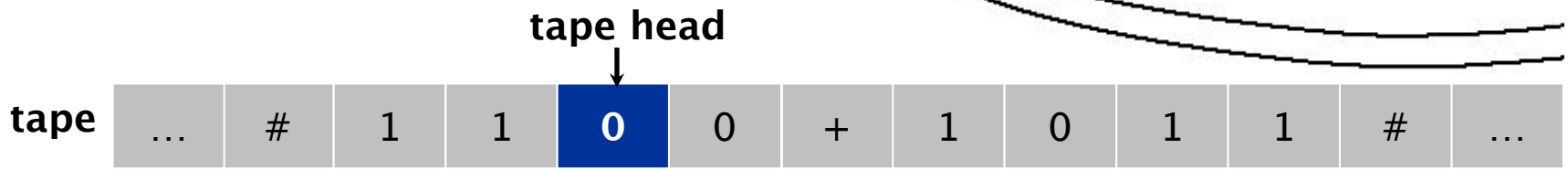
- Universal Turing machines
  - An abstract general purpose computer

Universal Turing Machines

- Tape
  - Stores input, output, and intermediate results.
  - One arbitrarily long strip, divided into cells.
  - Finite alphabet of symbols.

- Tape head
  - Points to one cell of tape.
  - Reads a symbol from active cell.
  - Writes a symbol to active cell.
  - Moves one cell at a time.





- Is there a more powerful model of computation? No!

Most important scientific result of 20th century?

Questions About Computation

- What is a general-purpose computer?
- Are there limits on the power of digital computers?
- Are there limits on the power of machines we can build?


David Hilbert


Kurt Gödel


Alan Turing


Alonzo Church


John von Neumann

Church-Turing Thesis (1936)

Turing machines can compute any function that can be computed by a physically harnessable process of the natural world.

- **Remark.** "Thesis" and not a mathematical theorem because it's a statement about the physical world and not subject to proof.
- Use simulation to prove models equivalent.
  - Android simulator on iPhone.
  - iPhone simulator on Android.
- **Implications.**
  - No need to seek more powerful machines or languages.
  - Enables rigorous study of computation (in this universe).
- **Bottom line.** Turing machine is a simple and universal model of computation.

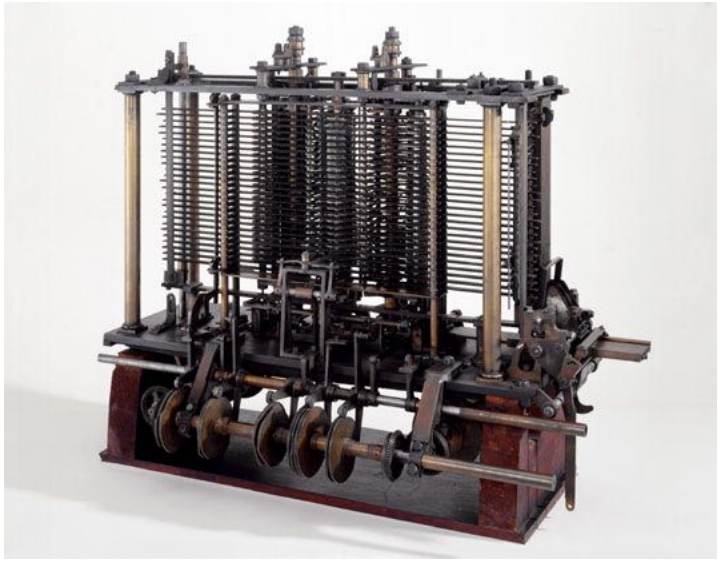
Church-Turing Thesis: Evidence

- 8 decades without a counterexample.
- Many, many models of computation that turned out to be equivalent.

model of computation	description
enhanced Turing machines	multiple heads, multiple tapes, 2D tape, nondeterminism
untyped lambda calculus	method to define and manipulate functions
recursive functions	functions dealing with computation on integers
unrestricted grammars	iterative string replacement rules used by linguists
extended L-systems	parallel string replacement rules that model plant growth
programming languages	Java, C, C++, Perl, Python, PHP, Lisp, PostScript, Excel
random access machines	registers plus main memory, e.g., TOY, Pentium
cellular automata	cells which change state based on local interactions
quantum computer	compute using superposition of quantum states
DNA computer	compute using biological operations on DNA

Babbage's Analytical Engine (1834, 1836)

- Designed around 1834 to 1836
  - was to be a universal machine capable of any mathematical computation
  - embodies many elements of today's digital computer
  - a control unit with moveable sprockets on a cylinder that could be modified
  - separated the arithmetic operations (done by the mill) from the storage of numbers (kept in the store)
    - store had 1000 registers of 50 digits each
  - Babbage incorporated using punched cards for input
    - idea came from Jacquard loom
- Never built by Babbage due to lack of funds and his eventual death in 1871



Adopted from: T. Cortina

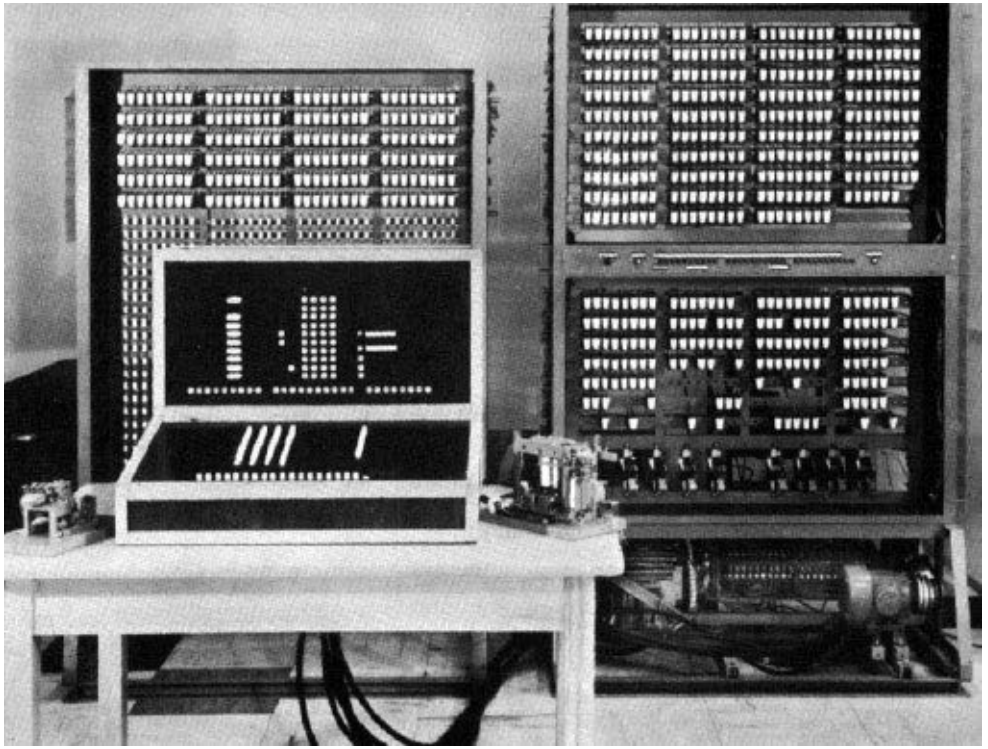
Ada Lovelace

- 1815-1852
- Daughter of poet Lord Byron
- Translated Luigi Menabrea's article on Babbage's Analytical Engine to English
  - Quadrupled its length by adding lengthy notes and detailed mathematical explanations
- Referred to as the world's first programmer
  - Described how the machine might be configured (programmed) to solve a variety of problems.

Slide credit: G. Kesden

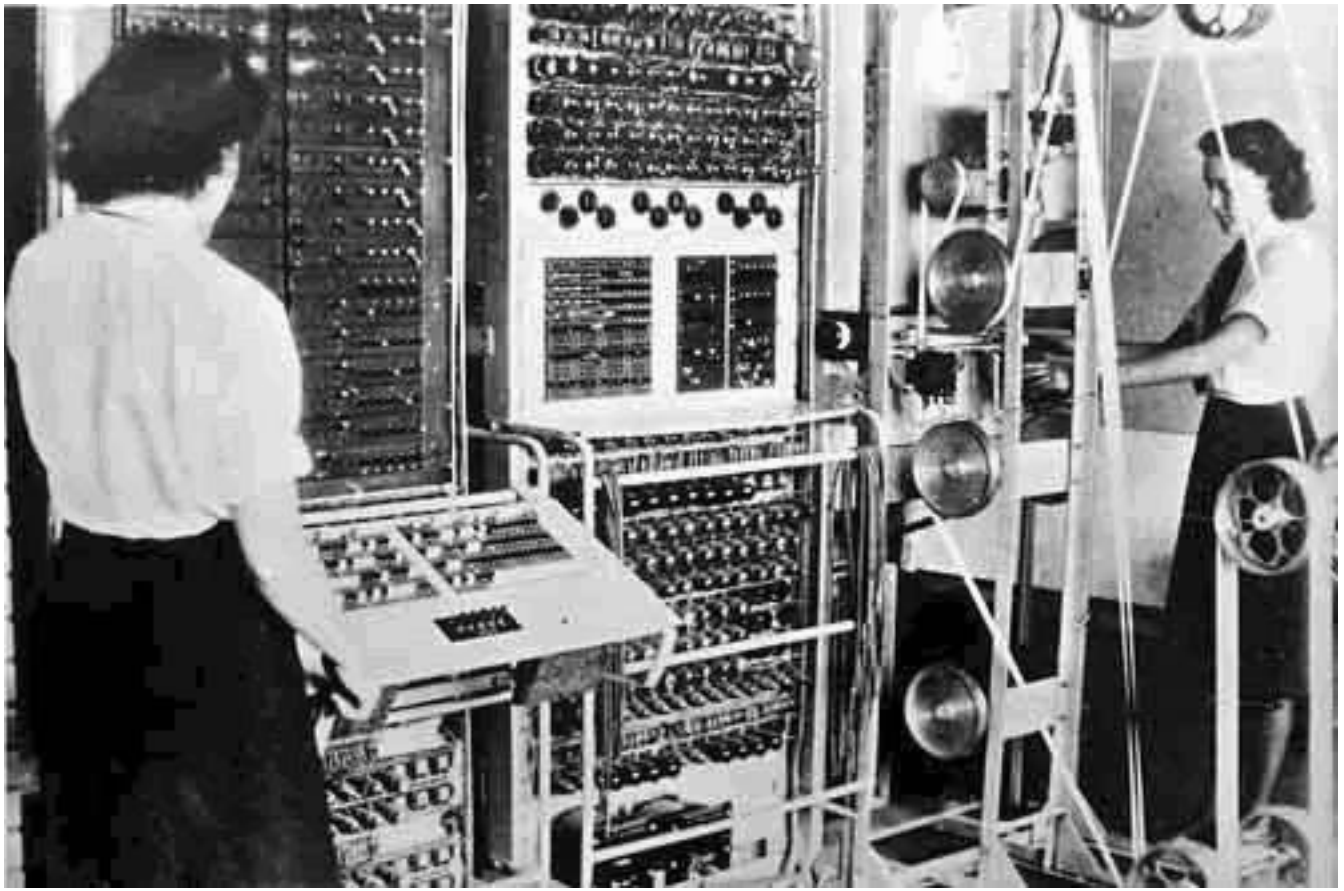
The Zuse Z3 Computer (1941)

- Konrad Zuse, 1941
- The original Z3 was destroyed in a bombing raid of Berlin in 1943.
- Zuse later supervised a reconstruction of the Z3 in the 1960s (currently on display at the Deutsches Museum in Munich)



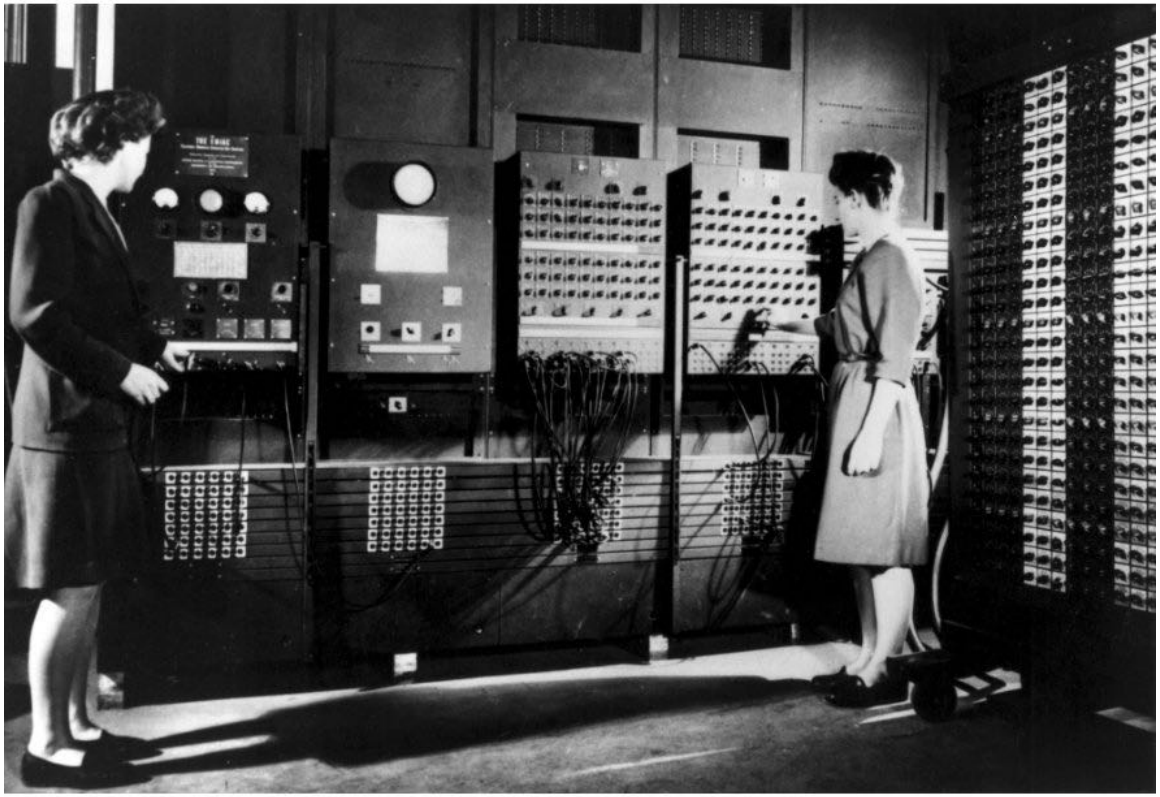
Colossus Mark 1 (UK,1944)

- The world's first electronic digital computer with programmability.



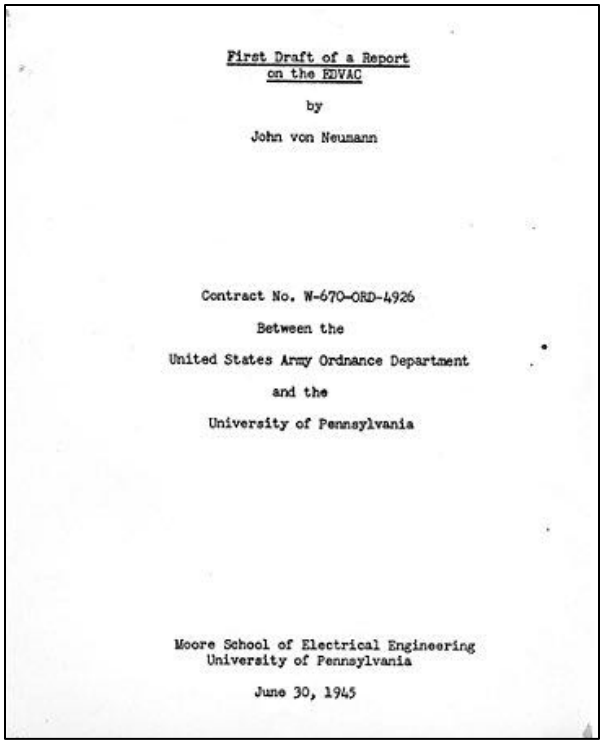
ENIAC (Mauchly and Eckert, USA, 1946)

- The first large-scale general-purpose electronic computer without any mechanical parts.
- Designed to calculate artillery firing tables for the United States Army's Ballistic Research Laboratory



EDVAC (von Neuman, USA, 1951)

- Unlike the ENIAC, it uses binary rather than decimal numbering system
- Instructions were stored in memory sequentially with their data
- Instructions were executed sequentially except where a conditional instruction would cause a jump to an instruction someplace other than the next instruction



First Draft of a Report
on the EDVAC

by

John von Neumann

Contract No. W-670-ORD-4926

Between the

United States Army Ordnance Department

and the

University of Pennsylvania

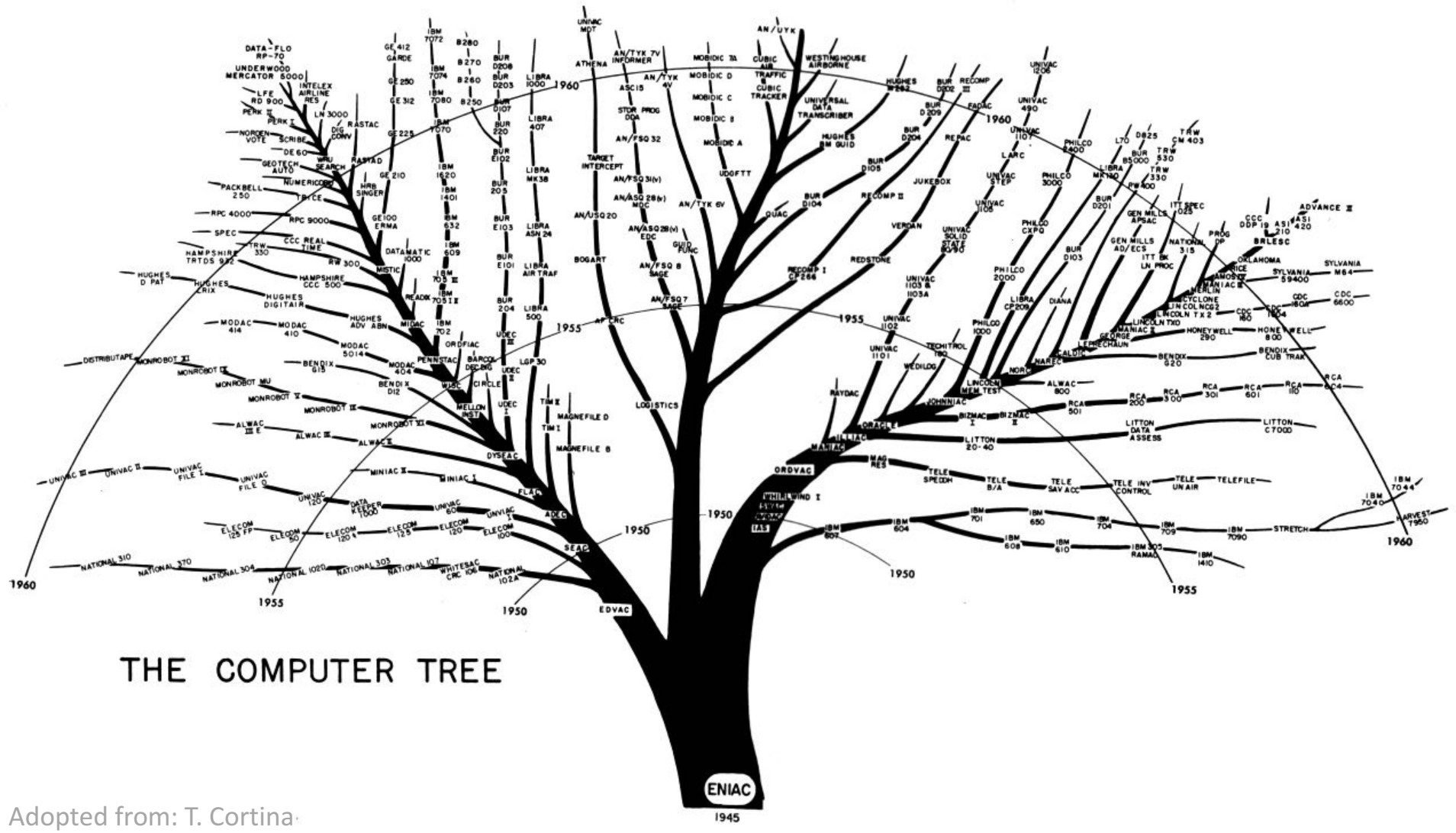
Moore School of Electrical Engineering
University of Pennsylvania

June 30, 1945

Adopted from G. Kesden

The Computer Tree

- http://ftp.arl.mil/~mike/comphist/61ordnance/chap7.html



Adopted from: T. Cortina

Summary

- **What is computation?**
  - What is knowledge?
  - What is a computer?
  - What is a program?
  - History of computing

The Birth of the Computer

- A TED talk given by George Dyson

http://www.ted.com/talks/george_dyson_at_the_birth_of_the_computer.html