MATH105A
Introduction To Computer Science

Szu-Chi Chung

Department of Applied Mathematics, National Sun Yat-sen University

Lectures

- Class hours: Fri. (9:10-12:00)
  - Classroom: 理 SC 2004
- Lecture: Szu-Chi Chung (鍾思齊)
  - Office: 理 SC 2002-4
  - Office hours: Mon. 16:00~18:00 and Wed. 16:00~18:00
- T.A.: 錢映伶
  - Office: 理SC 2003-3
  - Tutorial hours: Fri. 12:00~13:00 (at 理 SC 2004)
  - TA hour: Thur. 11:00~13:00 (at 理SC 2003-3)
- Facebook
- Course website
  - https://phonchi.github.io/nsysu-math105A/

Textbook and requirement

- Textbook: *Foundations of Computer Science, 4th Edition*
  - Authors: Behrouz A. Forouzan
  - https://www.cengageasia.com/TitleDetails/isbn/9781473751040
  - https://www.tsanghai.com.tw/book_detail.php?c=173&no=3993#p=1

- For the odd number of exercises of each chapter, the solution is at the companion website
  - https://www.cengage.com/cgi-wadsworth/course_products_wp.pl?fid=M20b&product_isbn_issn=9781473751040

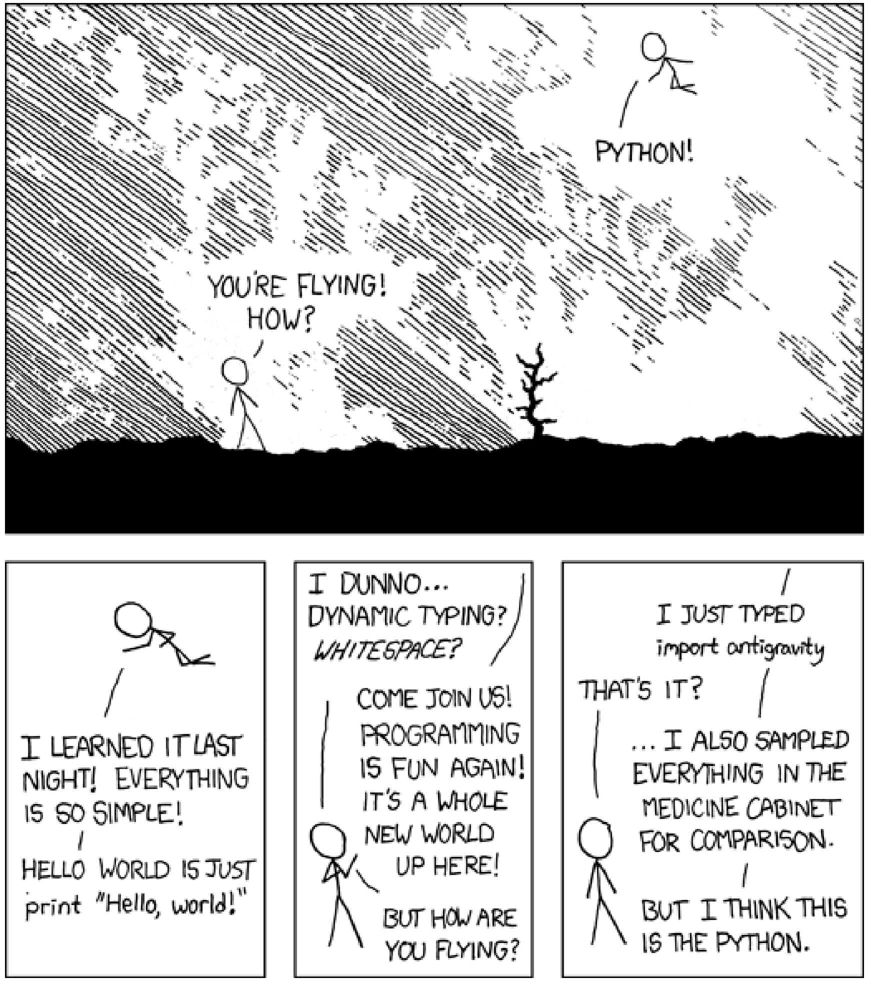
- The assignment and related material will be available on the course webpage

Grading policy

- Grading
  - Homework 24% (8~10 assignments, both conceptual and coding part (Python))
  - Participants: 6% (participates at least 10 times can get the full score)
  - Take home Quiz: 10% (2 times)
  - Midterm exam 30%
  - Final project 30%
- Midterm (conceptual):
  - It will be held on 2022/11/04 at 理 SC 2004
- Final (both conceptual and coding part):
  - It will be held on 2022/12/30 at 理 SC 2004

Grading policy

- Programming language: Python
  - You are asked to use python to implement a part of the assignment
  - Since it is one of the most popular languages and has a vibrant community support
  - It is free and easy to learn
- Python
  - **[Learn X in Y minutes]**
  - [Python for Everybody]



https://xkcd.com/353/

What we are going to study in this semester

- Introduction
- Data representation and operation
  - Number Systems
  - Data Storage
  - Operations on Data
- Computer hardware
  - Computer organization
  - Computer network and internet
- Computer software
  - Operating system
  - Programming Languages (Python)
  - Algorithms
- Data organization and abstraction
  - File Structure
  - Data Structure
  - Abstract Data Type
- Advance topics
  - Security
  - Artificial intelligence (Recorded lecture)
- Not covered
  - Software engineering
  - Databases
  - Data Compression
  - Theory of Computation
  - Social media and social Issues

Relate to other courses

- Related courses
  - Computer programming
  - Data structures
  - Algorithms
- Other courses
  - Computer organization
  - Computer network
  - Operating systems
  - Programming
  - Software engineering
  - Databases
  - Information security/Machine learning/Artificial intelligence

Introduction

Szu-Chi Chung

Department of Applied Mathematics, National Sun Yat-sen University

Introduction

- We define computer science as *'issues related to the computer'*
  - First tries to find out what a computer is
  - Investigates other issues directly related to computers.
- We look first at the Turing model
  - A mathematical and philosophical definition of computation.
- We then show how today's computers are based on the
  - von Neumann model

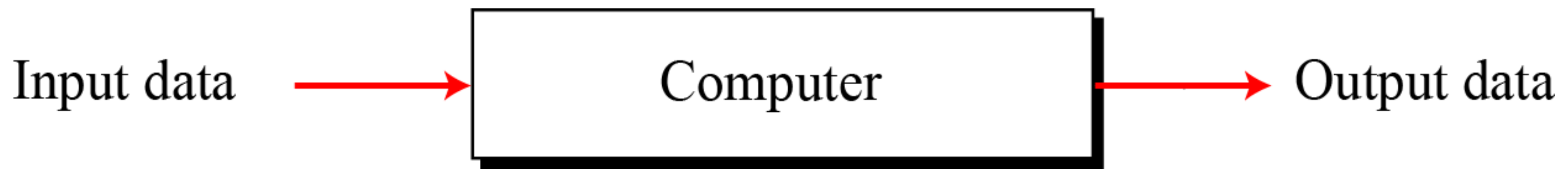
Turing model

- The idea of a universal computational device was first described by Alan Turing in 1936
  - He proposed that all computation could be performed by a special kind of machine, now called a *Turing machine*
- Although Turing presented a mathematical description of such a machine, he was more interested in the philosophical definition of computation than in building the actual machine

Data processors

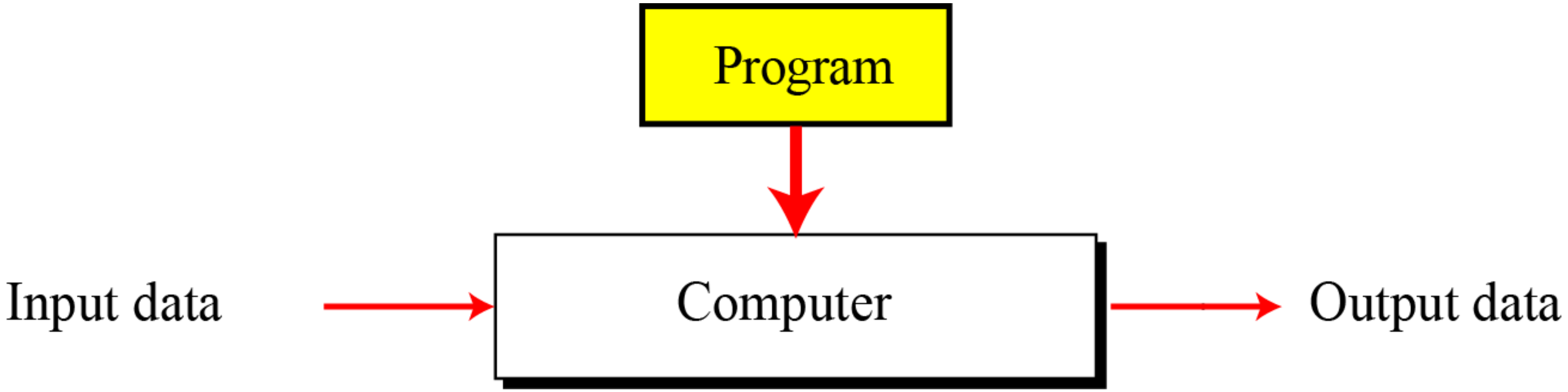
- A computer acts as a black box that accepts input data, processes the data, and creates output data
  - Too general. Is a pocket calculator a computer?
  - This model could represent a specific-purpose computer like the controller that used to control the fuel usage in a car

- Computers, as the term is used today, are *general-purpose* machines



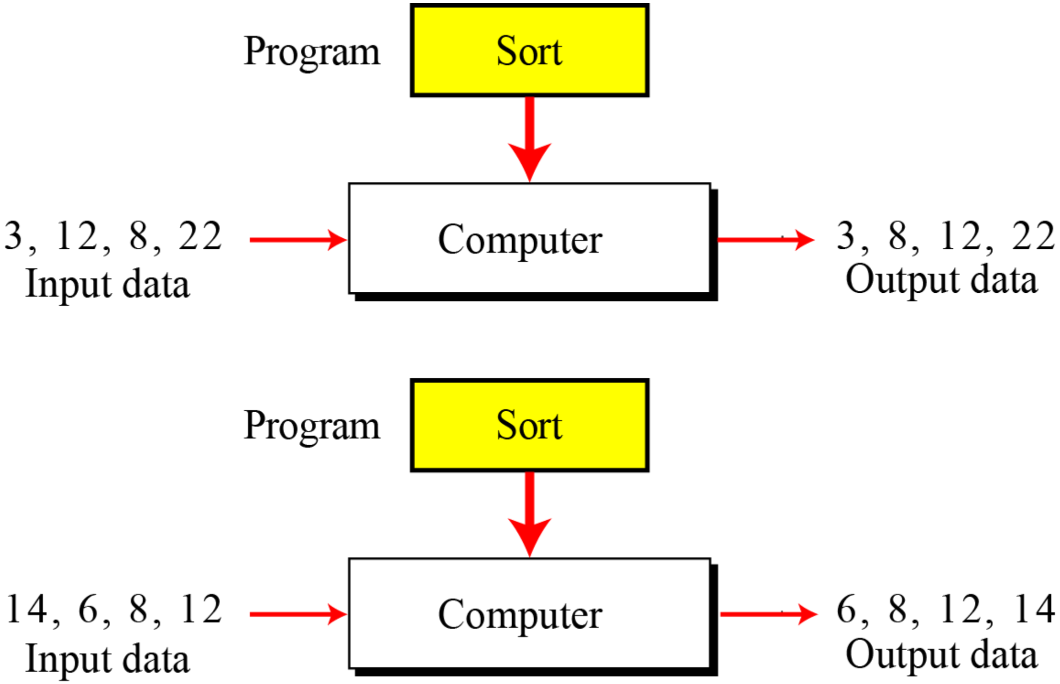
Turing model - Programmable data processors

- The *Turing model* is a better model for a general-purpose computer
  - A *program* is a set of instructions that tells the computer what to do with data
- The *output data* now depends on the combination of the *input data* and the program

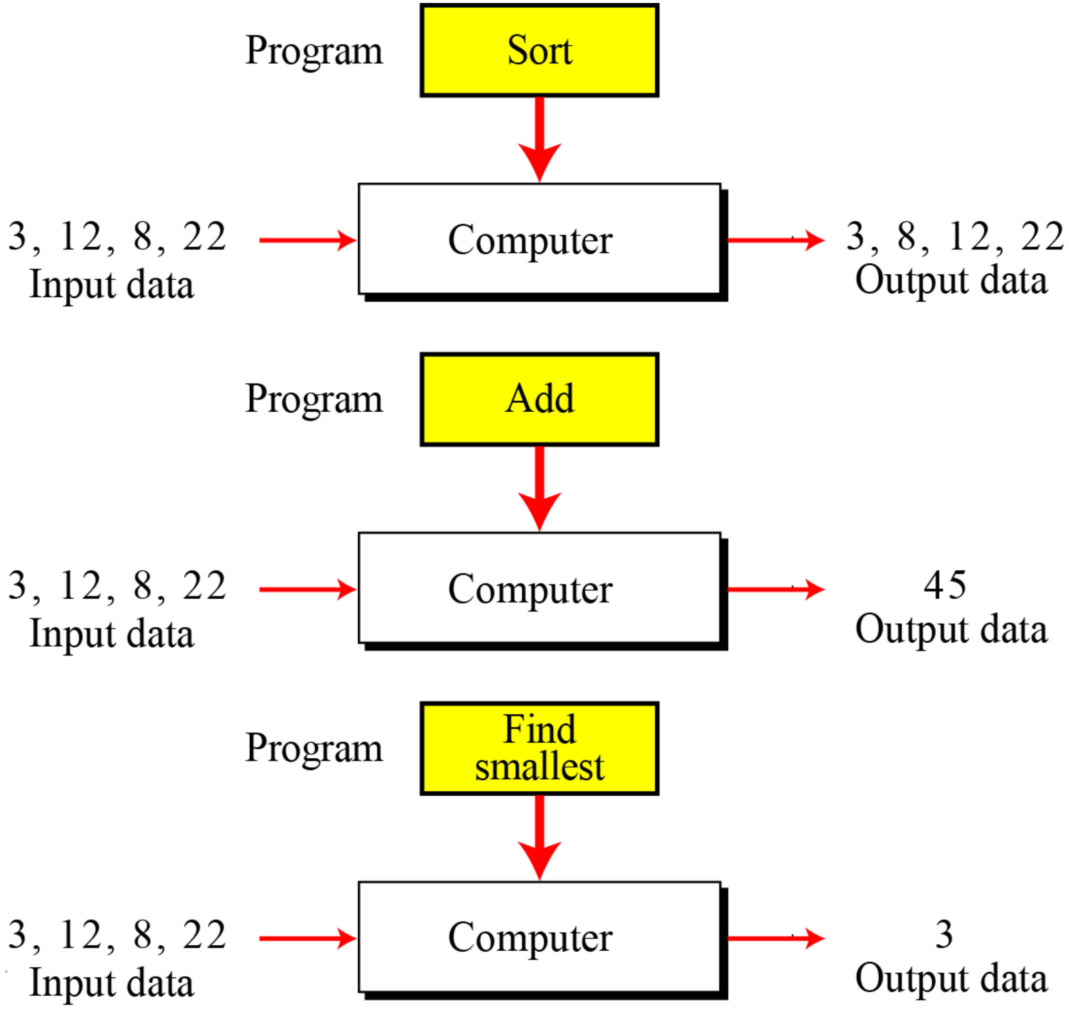


Turing model - Programmable data processors

- Same program, different input data



- Same input data, different programs

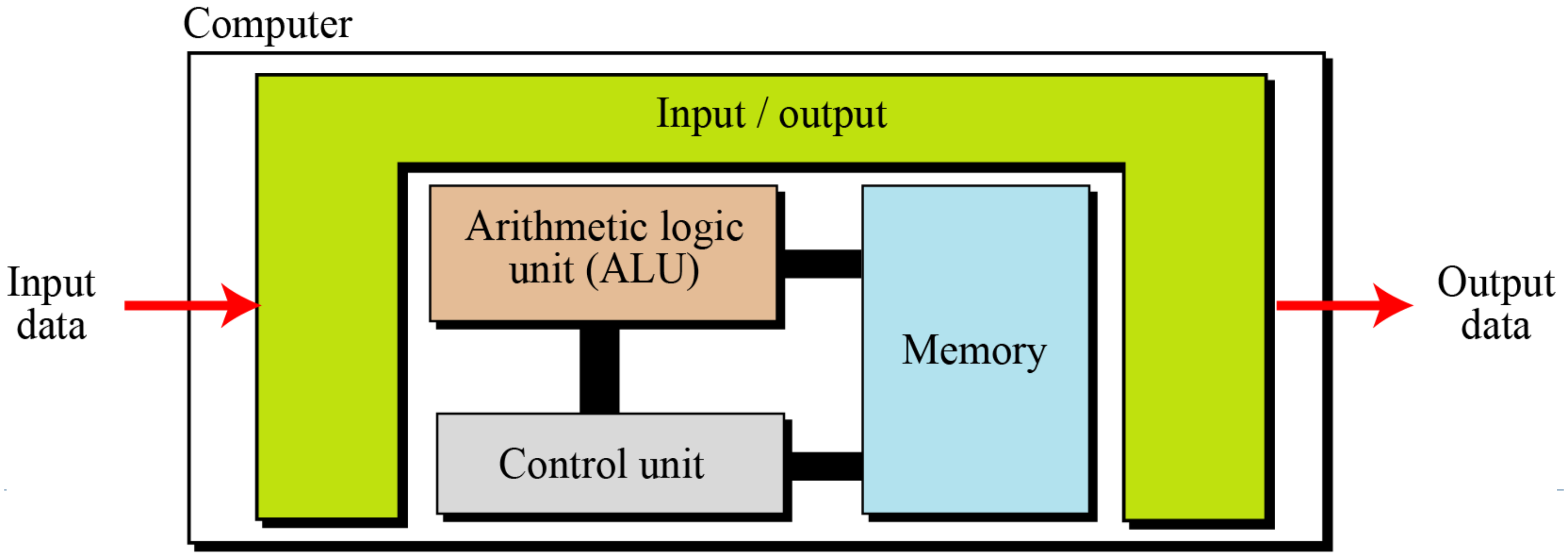


The universal Turing machine

- A *universal Turing machine* is a machine that can do any computation if the appropriate program is provided
  - The first description of the modern computer
  - It can be proved that a very powerful computer and a universal Turing machine can compute the same thing
  - In fact, a universal Turing machine is capable of computing anything that is computable

von Neumann model

- Computers built on the Turing universal machine store data in their memory
  - John von Neumann proposed that, since programs and data are logically the same, programs should also be stored in the memory of a computer
- Computers built on the von Neumann model divide the computer hardware into four subsystems
  - *Memory, arithmetic logic unit, control unit, and input/output*



von Neumann model

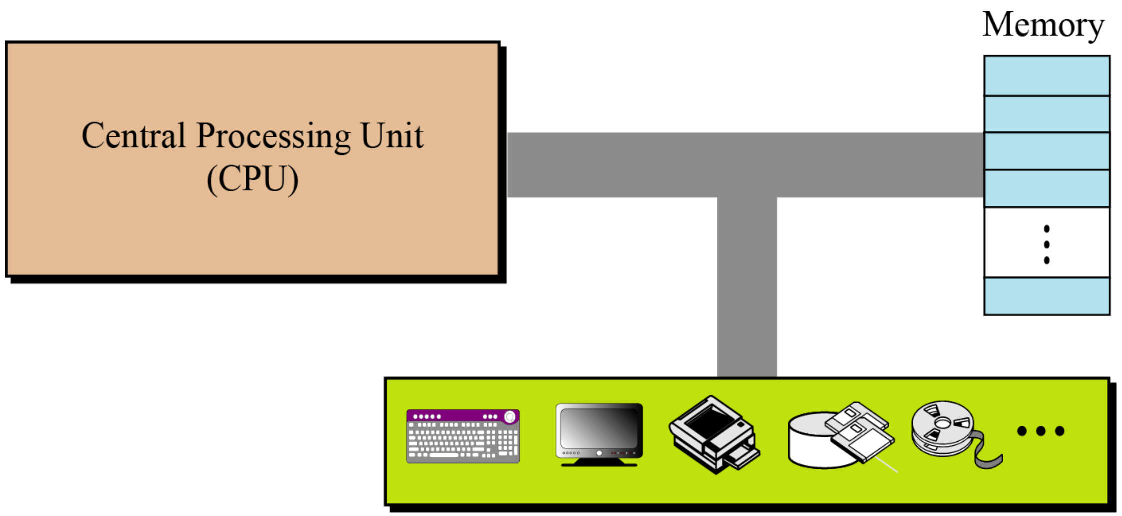
- Memory
  - This is where programs and data are stored during processing
- Arithmetic logic unit (ALU)
  - This is where calculation and logical operations take place. For a computer to act as a data processor, it must be able to do arithmetic and logical operations on data
- Control unit
  - It controls the operations of the memory, ALU, and the input/output subsystem
- Input/output
  - The input subsystem accepts input data and the program from outside the computer, while the output subsystem sends the result of processing to the outside world

von Neumann model - The stored program concept

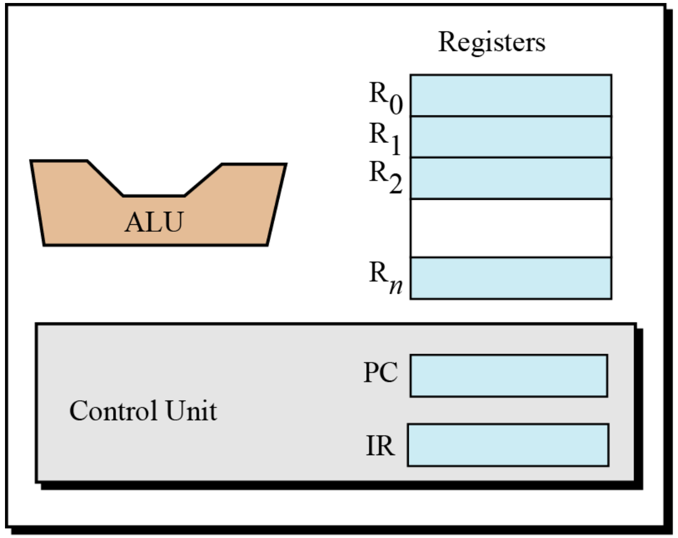
- The program must be stored in memory is a new concept
  - The programs were implemented by manipulating a set of switches or by changing the wiring system in the early architecture
- This implies that both the data and programs should have the same format
  - They are stored as *binary* patterns in memory—a sequence of 0s and 1s
- A program is made of a finite number of *instructions*
  - In this model, the control unit fetches one instruction from memory, decodes it, then executes it in a *sequential* manner

Computer components

- We can think of a computer as being made up of three components: *computer hardware, data, and computer software*
- Computer hardware
  - Computer hardware today has four components under the von Neumann model, although we can have different types of memory, different types of input/output subsystems



Input / output subsystem



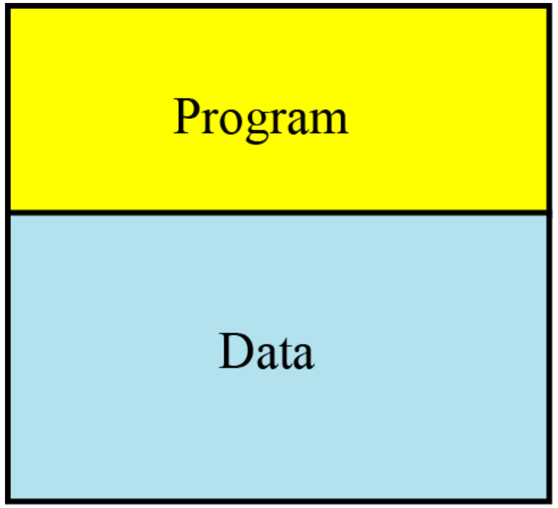
Central Processing Unit (CPU)

Computer components - data

- Storing data
  - If a computer is an electronic device, the best way to store data is in the form of an electrical signal, specifically its presence or absence
  - This implies that a computer can store data in one of *two* states
  - The numbering system uses digits that take one of ten states (0 to 9) and other types of data (text, image, audio, video) may involve more states which need to be transformed
- Organizing data
  - Although data should be stored only in one form inside a computer, we can organize our data into different entities and formats before storing them
  - Data is organized into small units, small units are organized into larger units, and so on

Computer components – computer software

- In the von Neumann model programs are stored in the computer's memory. Not only do we need memory to hold data, but we also need memory to hold the program
  - Although early computers did not store the program in the computer's memory, they did use the concept of programs. Programming those early computers meant changing the wiring systems or turning a set of switches on or off



Memory

Computer components – computer software

- The program must consist of a sequence of instructions.
  - Each instruction operates on one or more data items. Thus, an instruction can change the effect of a previous instruction
  - The instruction enhances the *reusability*
    - By defining different instructions that can be used by computers. A programmer can then combine these instructions to make any number of programs

1. Input the first number into memory.
2. Input the second number into memory.
3. Add the two together and store the result in memory.
4. Output the result.

Program

Computer components – computer software

- Algorithms
  - A programmer must not only learn the task performed by each instruction but also learn how to combine these instructions to do a particular task.
  - A programmer must first, solve the problem in a step-by-step manner, then try to find the appropriate instructions to implement those steps which are the field of algorithms
- Languages
  - In the early stage, programmers wrote instructions (*machine language*) to solve a problem
  - As programs became larger, the idea of using symbols to represent binary patterns leads to the development of *computer languages*

Computer components – computer software

- Software engineering
  - Something that was not defined in the von Neumann model is software engineering, which is the design and writing of *structured programs*
- Operating systems
  - There was a series of instructions common to all programs!
  - An operating system originally worked as a manager to facilitate *access* to the computer's components by a program

History - Mechanical machines (before 1930)

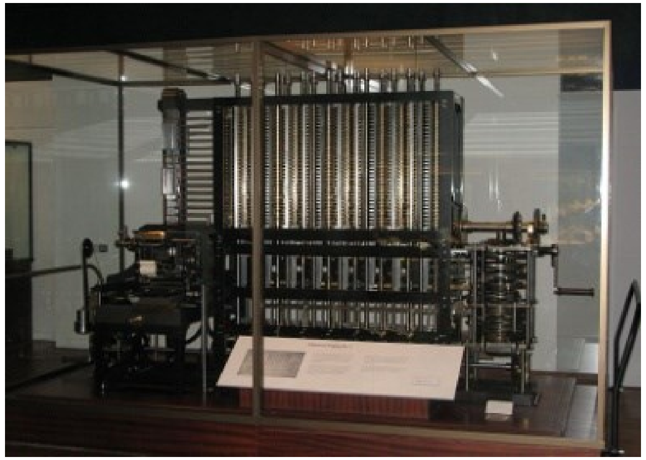
- In the 17th century, Blaise Pascal, a French mathematician and philosopher, invented *Pascaline* – the first mechanical calculator
- In the late 17th century, German mathematician Gottfried Leibnitz invented *Leibnitz' Wheel* – a calculator that could do multiplication and division
- The first machine that used the idea of storage and programming (punched cards) was the *Jacquard loom* at the beginning of the 19th century

https://en.wikipedia.org/wiki/Pascal%27s_calculator

https://en.wikipedia.org/wiki/Jacquard_machine

History - Mechanical machines (before 1930)

- In 1823, Charles Babbage invented the *Difference Engine*. Later, he invented the machine called the *Analytical Engine* which parallels the idea of modern computers
- In 1890, Herman Hollerith, designed and built a programmer machine that could automatically read, tally, and sort data stored on punched cards



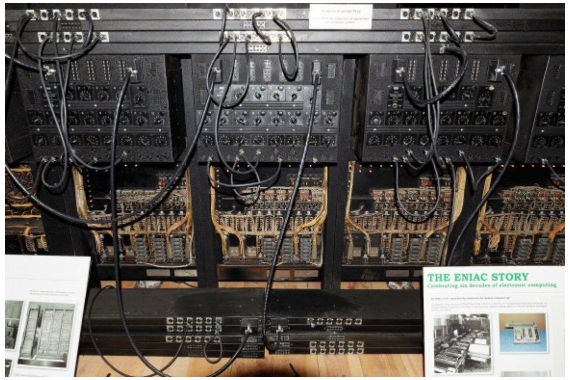
https://en.wikipedia.org/wiki/Difference_engine

https://en.wikipedia.org/wiki/Herman_Hollerith

History - birth of electronic computers (1930–1950)

- The early computers of this period did not store the program in memory — all were programmed externally
  - ABC - The first special-purpose computer that encoded information electrically
  - Z1 - a general-purpose machine
  - Mark I – a computer used both electrical and mechanical components
  - Colossus – design by Alan Turing to break German Enigma code
  - ENIAC - The first general-purpose, totally electronic computer

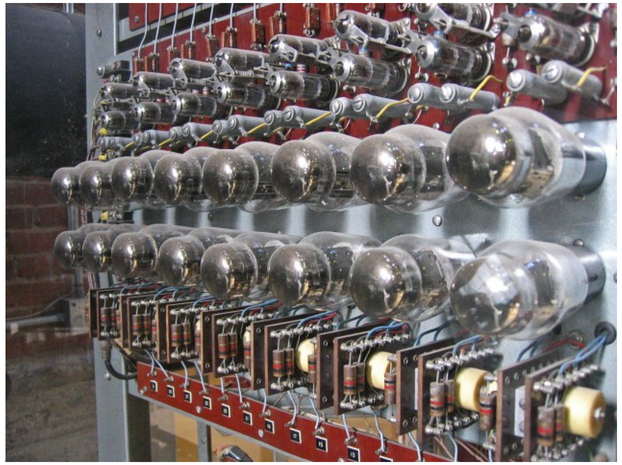
https://en.wikipedia.org/wiki/Harvard_Mark_I



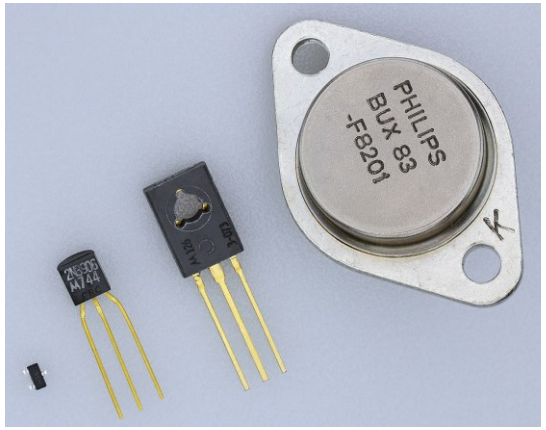
https://en.wikipedia.org/wiki/ENIAC

History - Computer generations (1950–present)

- The first computer based on von Neumann's ideas was called EDVAC
- First generation (roughly 1950–1959)
  - Characterized by the emergence of commercial computers that used vacuum tubes as electronic switches
- Second generation (roughly 1959–1965)
  - Used transistors instead of vacuum tubes. Two high-level programming languages, FORTRAN and COBOL invented and made programming easier



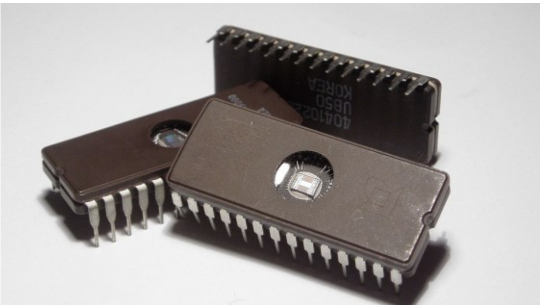
https://en.wikipedia.org/wiki/Vacuum_tube



https://en.wikipedia.org/wiki/Transistor

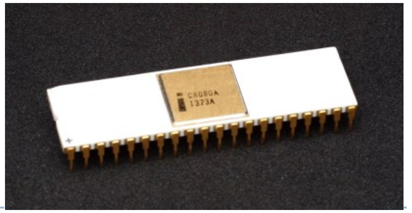
History - Computer generations (1950–present)

- Third generation (roughly from 1965 to 1975)
  - The invention of the integrated circuit reduced the cost and size. Minicomputers appeared on the market. Canned programs, popularly known as software packages, became available
- Fourth generation (approximately 1975–1985)
  - We saw the appearance of microcomputers. The first desktop calculator, the Altair 8800, became available in 1975. This generation also saw the emergence of computer networks



https://en.wikipedia.org/wiki/Integrated_circuit

https://en.wikipedia.org/wiki/Altair_8800



https://en.wikipedia.org/wiki/Intel_8080

https://en.wikipedia.org/wiki/IBM_Personal_Computer

History - Computer generations (1950–present)

- Fifth generation (1985-)
  - It has witnessed the appearance of laptop, tablet computers and smart phone, improvements in secondary storage media (CDROM, DVD), the use of multimedia et al.

https://en.wikipedia.org/wiki/Epson_HX-20

https://en.wikipedia.org/wiki/Laptop

https://en.wikipedia.org/wiki/Tablet_computer

https://en.wikipedia.org/wiki/Smartphone

Computer Science as a discipline

- With the invention of computers, a new discipline *computer science* has evolved and has been divided into several areas
- Systems areas
  - Directly related to the creation of hardware and software
  - Computer architecture, computer networking, security issues, operating systems, algorithms, programming languages and software engineering
- Application areas
  - Related to the use of computers
  - Databases and artificial intelligence