

# C Programming

## Lecture 1: An Introduction and Overview on C



Lecturer: Dr. Wan-Lei Zhao  
*Autumn Semester 2022*

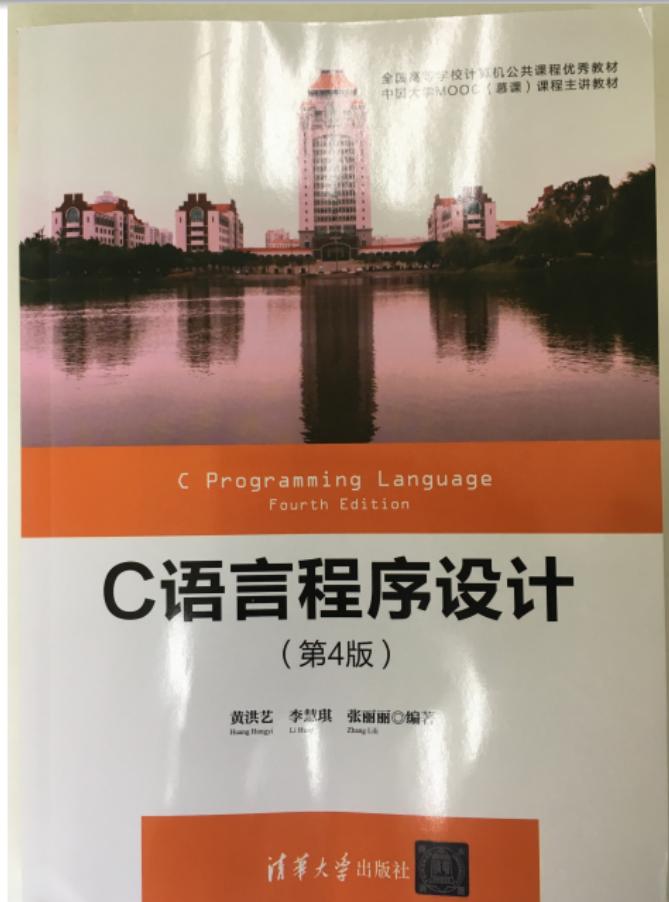
# Outline

- 1 Syllabus
- 2 All about Computer
- 3 Programming
- 4 Basics about C Programming

# Syllabus

- ① Primitive Data Types and Operations
  - ② Sequential Control
  - ③ Selection Control clause: `if-else` and `switch`
  - ④ Loops Control clause: `while`, `do-while` and `for`
  - ⑤ Functions: declaration, definition and calling
  - ⑥ Pre-compilation Command/Macros: `#ifdef`
  - ⑦ Array: declaration, definition and calling
  - ⑧ Structures: `struct` and `union`
  - ⑨ Pointers
  - ⑩ File Operations: read and write
- Performance Evaluation
    - $\text{Final score} = 10\% \times \text{Exerc.} + 30\% \times \text{Quiz.} + 10\% \times \text{Att.} + 50\% \times \text{Exam}$

# Textbook



# Arrangement of this course

- 16 weeks×2 hours classes
- 8 weeks×2 hours labs
  - TA and I will be in the lab
- Middle-term exam
- Doing final exam, both are held in the lab
  - Multiple choices
  - Correct codes
  - 3-4 coding problems
- **No cheating and no bargaining!**
- If you attend all my classes
- I ensure that you can learn a lot:)



(a) Yi-Bo Miao  
(B308, 8:00-9:50)



(b) Jie-Feng Wang  
(B311, 10:10-11:50)

- ① Annoucements
- ② Assistance in the Lab
- ③ Answer your questions

- PTA: <https://pintia.cn/>



- ① Register with your email account
- ② You can type your codes, submit and compile
- ③ You should print out the exact answer

# Outline

1 Syllabus

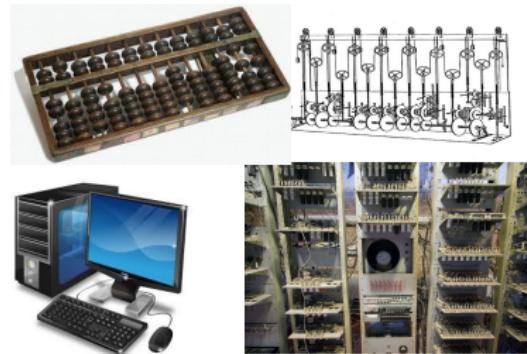
2 All about Computer

3 Programming

4 Basics about C Programming

# About Computer (1)

- What is computer?
  - Machine for computation
  - Essentially, no big difference from abacus
  - In our history, we have several kinds of machines used for computing
    - Abacus
    - Difference engine
    - Tide-predicting machine

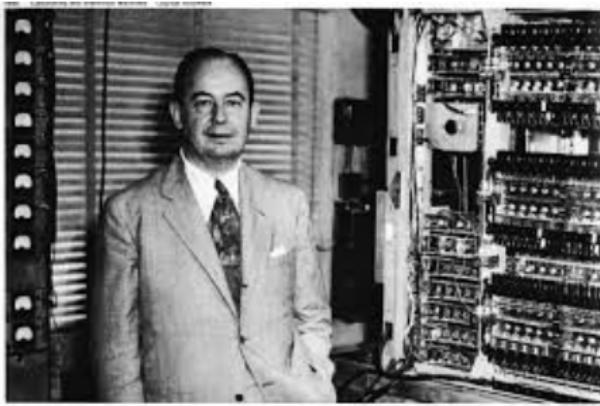


## About Computer (2): the model

- What is computing
  - Input data and needed operations
  - Output the answer
- This is actually the model proposed by **Alan Turing**



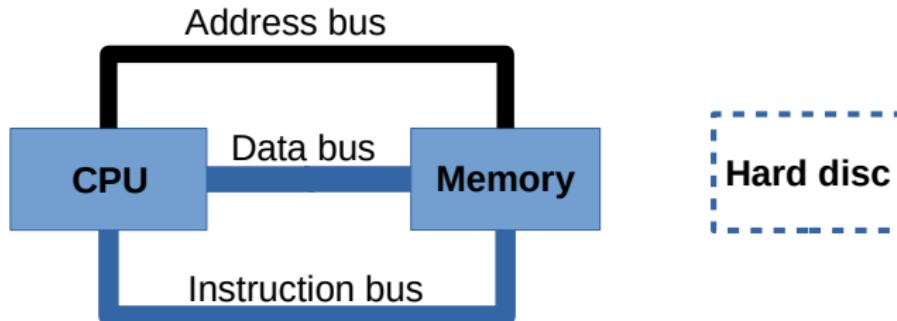
Alan Turing  
(1912-1954)



John Von Neumann  
(1903-1957)

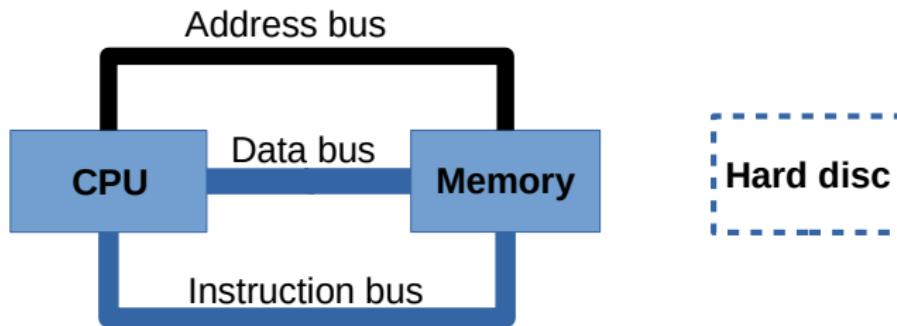
# About Computer (3): the framework

- Think aloud about the major components of a computer
  - CPU: central processing unit
  - Memory
  - Hard disc
  - Keyboard
  - graphics card+Monitor/screen
  - Music card+microphone+speaker
  - Mouse



# About Computer (4): the framework

- Think aloud about the major components of a computer
  - **CPU: central processing unit**
  - **Memory**
  - Hard disc
  - Keyboard
  - graphics card+Monitor/screen
  - Music card+microphone+speaker
  - Mouse

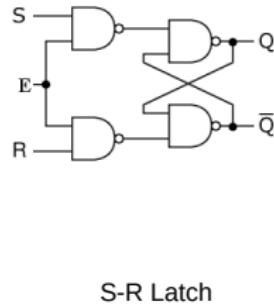
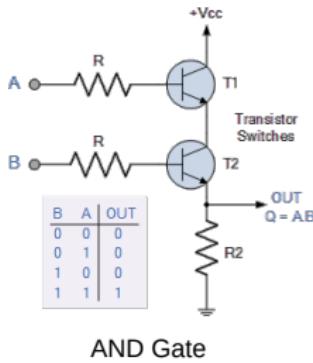


# About Computer (5): who is who



- How many of them you can finger out?

# About Computer (6): basic elements in Computer Chips



- Despite the high complexity of VLSIC (very large scale integrated circuits)
- Only two basic elements are there
- One is gate, responsible for operations, main components for CPU
- Another is latch, in charge of memory, main components for memory

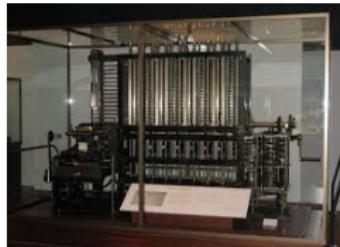
# Outline

- 1 Syllabus
- 2 All about Computer
- 3 Programming
- 4 Basics about C Programming

# Why programming? (1)



Charles Babbage  
(1792—1871)

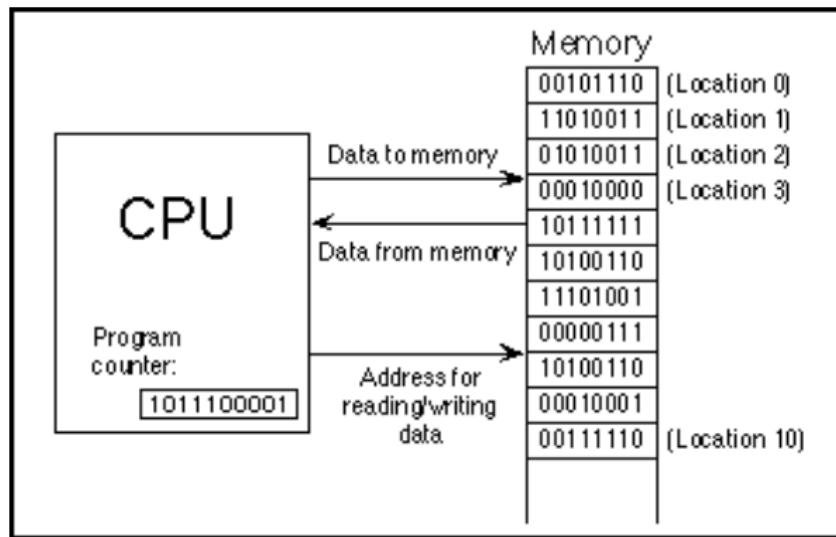


Mechanical computer



Ada Lovelace  
(1815-1852)

## Why programming? (2)



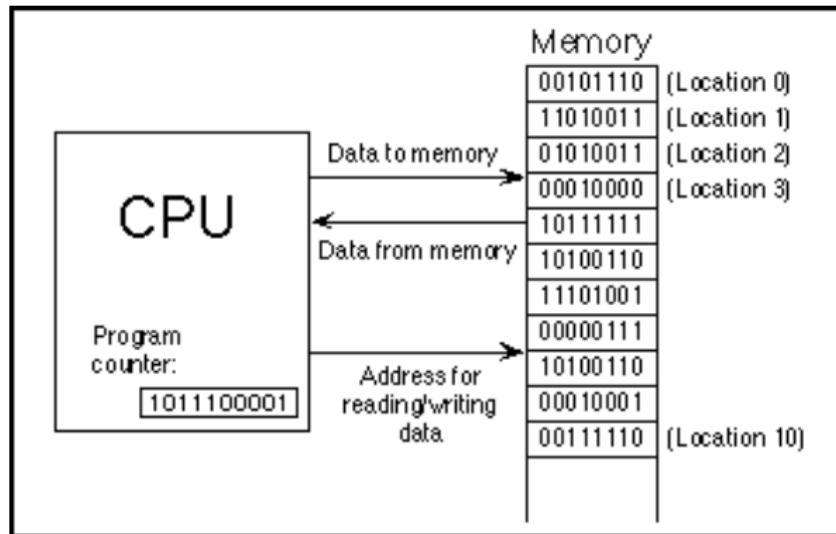
- Instructions and data fetch from memory to CPU for processing
- The results are returned back to memory

# Why High Level Programming Language? (1)



- Natural language is the media that we communicate with each other
- Computer language is the media that we communicate with computer
- We should use the language that computer could understand
- At least, we need an **interpreter/translator**

# Why High Level Programming Language? (2)



- Instructions are binary codes
- Machine only accepts/understands binary codes

# Why Programming Language? (3)

- ① 010101 0000 0011
- ② 010101 0001 0101
- ③ 101010 0000 0001
- ④ 010101 0000 1011

# Why Programming Language? (4)

- ① 010101 0000 0011
- ② 010101 0001 0101
- ③ 101010 0000 0001
- ④ 010101 0000 1011

- ① MOV D1 0011
- ② MOV D2 0101
- ③ ADD D1 D2
- ④ MOV D1 A1

- For the convenience of operation, binary instructions are denoted with readable symbols

# Why Programming Language? (5)

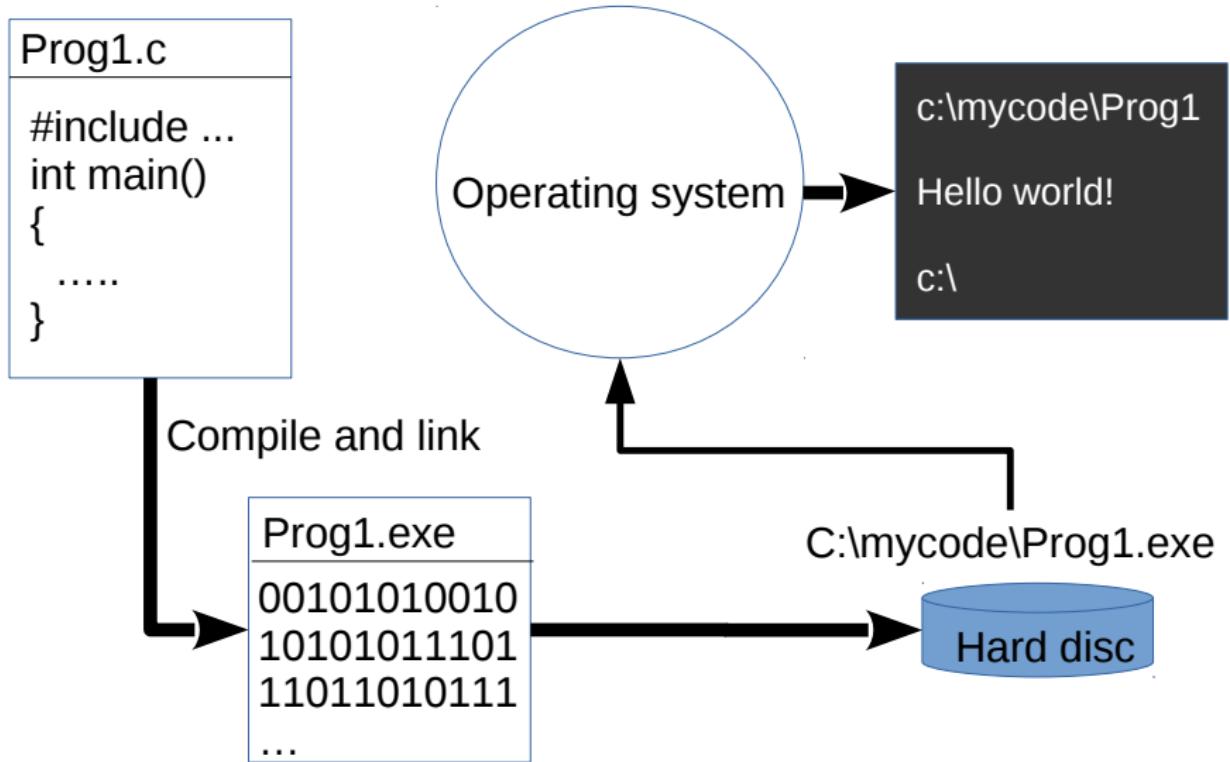
- Machine code
- ① 010101 0000 0011
- ② 010101 0001 0101
- ③ 101010 0000 0001
- ④ 010101 0000 1011
- Assembly
- ① MOV D1 0011
- ② MOV D2 0101
- ③ ADD D1 D2
- ④ MOV D1 A1
- High level language
- ①  $a=3+5;$

# Why Programming Language? (6)



- We write a **text** file in specified format (grammar)
- These are instructions that we basically understand
- The **translator** converts the text instructions into machine codes
- Machine then runs these binary codes one by one
- Different **translators** lead to different programming languages
- Which also regulate different grammars
- C is such kind of high level language

# The life-time of a computer program



# Outline

- 1 Syllabus
- 2 All about Computer
- 3 Programming
- 4 Basics about C Programming

# Brief History about C



Ken Thompson  
(1943 - )



Dennis M. Ritchie  
(1941 - 2011)

- C is born in AT&T Bell Labs along with UNIX
- The developer Dennis Ritchie and Ken Thompson were awarded with Turing Award
- C is simple, versatile and highly efficient (70% of assembly language efficiency)
- UNIX is one of the most stable operating systems so far developed

# Your first program in C (1)

```
1 #include <stdio.h>
2 int main()
3 { /*start of a block*/
4     printf("Hello_world!\n"); /*call function 'printf'*/
5     return 0;                /*return '0' back*/
6 } /*end of a block*/
```

- “#include <stdio.h>” states that we want to use **function** defined in “stdio.h”
- Our code is encapsulated in a function called “**main()**”
- In the main body of the function
- We output “Hello world!” to the screen
- “**printf()**” is a function **defined** in “stdio.h”
- **include**, **int** and **return** are reserved keywords

# Your first program in C (2)

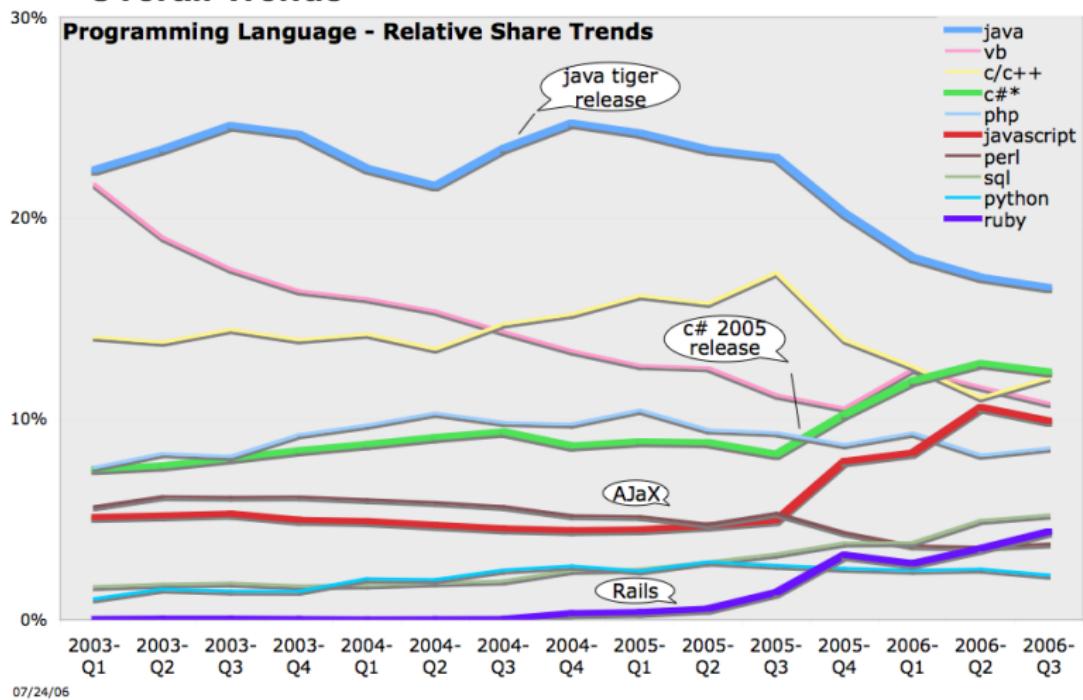
```
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello_world_1!\n");
5     printf("Hello_world_2!\n");
6     printf("Hello_world_3!\n");
7     return 0;
8 }
```

[Output]

```
1 Hello world 1!
2 Hello world 2!
3 Hello world 3!
```

- Codes are executed **from top to bottom**

# Popularity of C in recent decade



# Popularity of C in recent decade

