CS 1502H 程序设计思想与方法 (C++)
Principles and Methodology in Programming

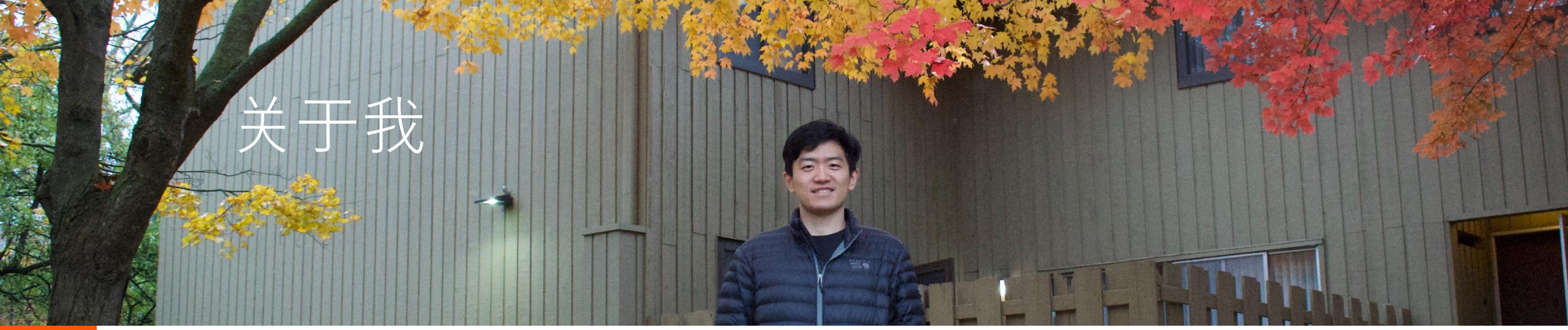
陈东尧
上海交通大学
2025年秋季

关于本课

- 这是基础的编程课程
- 必修
- 学时：80
- 学分：4.0
- 没有任何先修课程要求 

Outline

- 课程介绍
- 什么是计算机
  - 计算机的组成
- 什么是程序
- 程序设计流程概览



关于我

- 2013年从电院（自动化系）毕业，大约十年前我也上了程序设计课
- 2015年密西根大学安娜堡分校硕士，电子科学（EE）
- 2020年密西根大学安娜堡分校博士，计算机科学与工程（Computer Science and Engineering, CSE）
- 个人主页及实验室研究介绍
  - https://chendy.tech/

认识我们的助教

- 陈振宇

zhenyu.chen@sjtu.edu.cn

- 王纪科

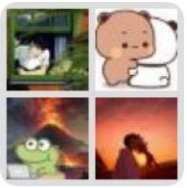
jikewang@sjtu.edu.cn

- 曾浩源

zenghaoyuan2020@sjtu.edu.cn

联系交流方式

- 邮件往往是最有效的方式：chendy@sjtu.edu.cn
- 关注canvas，群公告和作业信息会放在这里
- 我们的微信群，用于通知
- 我的办公室
  - 电信群楼，一号楼203



群聊: CS1501H-2025

该二维码7天内(9月22日前)有效，重新进入将更新

关于成绩

- **100 = 50+40+10**
  - “10”：课堂参与
  - “40”：平时作业
  - “50”：期末笔试
- 关于上机课（lab），我们会通过现场写程序的方式来：
  - 完成上机编程题目
  - 学习编程技能
  - 解答疑惑
- 集中答疑
  - [拟定]周一，16:00 – 17:00，电院一号楼，427会议室

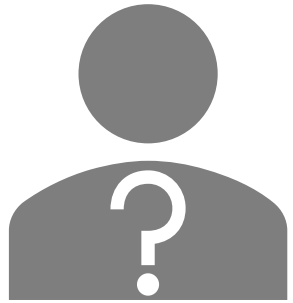
荣誉守则
Honor Code

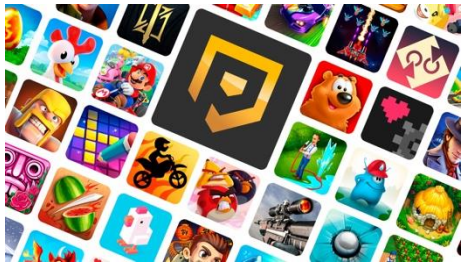
- 作弊是一件后果非常严重的事情

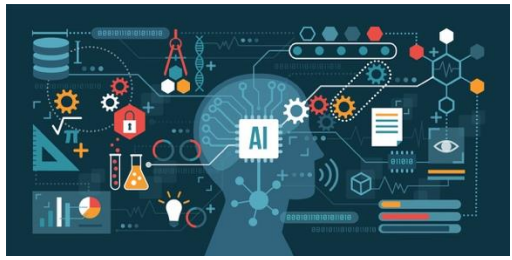
请一定不要借出或抄袭代码

关于你

- 你们的专业？
- 几个问题，有哪些同学：
  1. 了解如何使用电脑/手机？
  2. 听说过一些计算机程序语言？
  3. 尝试过自己写程序？







如果你也希望了解计算机和程序是怎么工作的，那么恭喜！

Outline

- 课程介绍
- 什么是计算机
  - 计算机的组成
- 什么是程序
- 程序设计流程概览

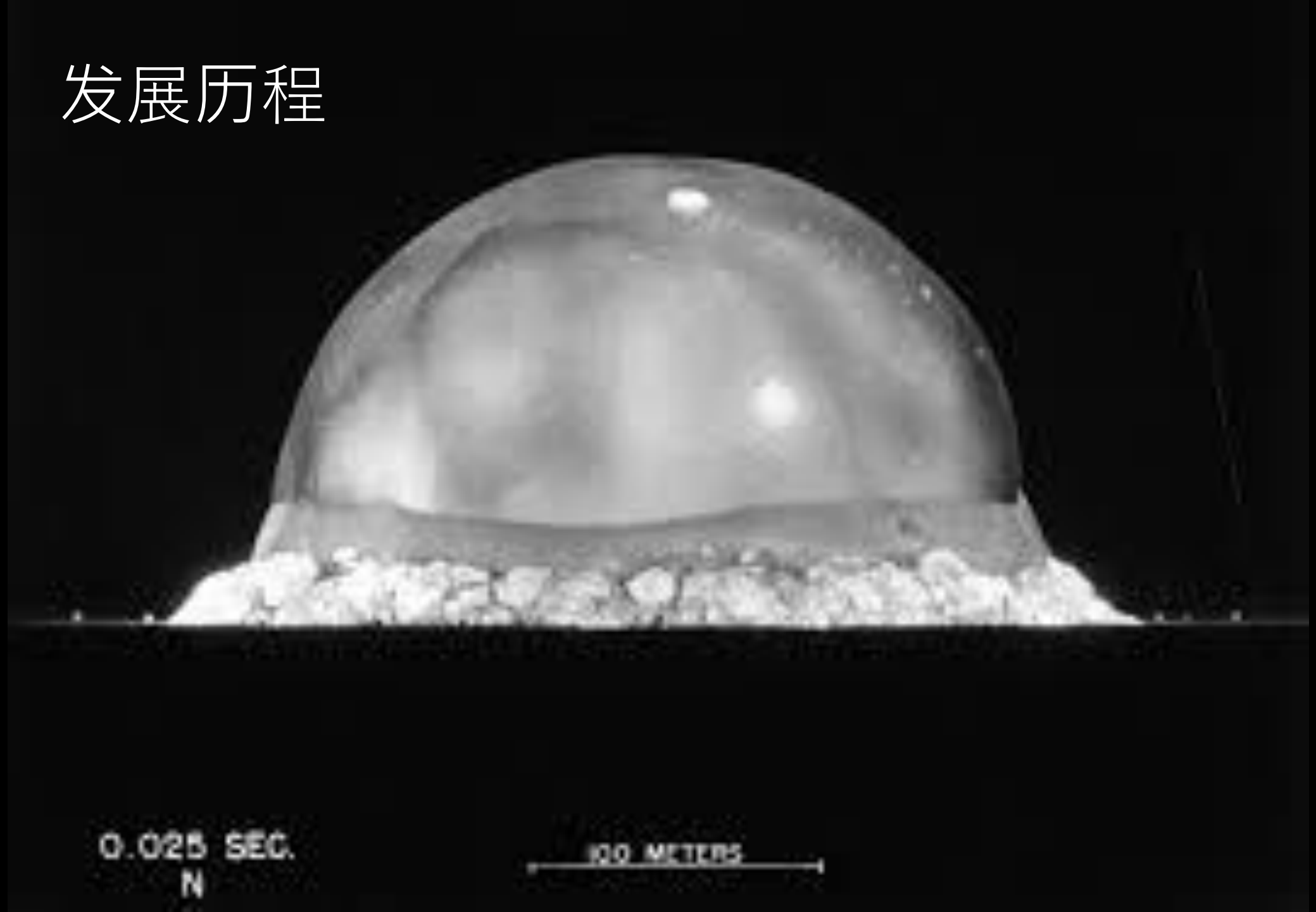
什么是计算机科学（Computer Science）？

- 计算机科学(CS)并非单纯研究计算机
  - 正如天文学不是研究天文望远镜 (E. W. Dijkstra)

- CS要回答的基本问题:
  哪些问题是可以计算的？如果可以，有多困难？

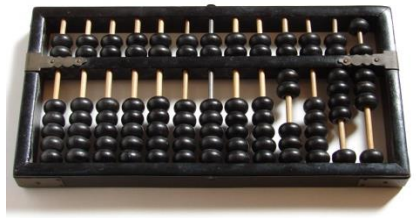
- 计算是指利用计算机执行程序来解决问题
  - 什么是计算机？
  - 什么是程序?

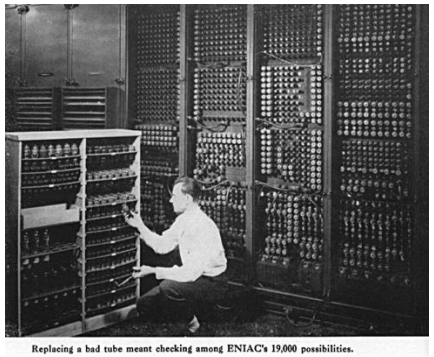
发展历程



发展历程

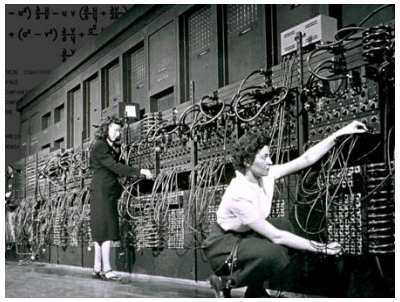
- 计算机简史





Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

- 程序员简史



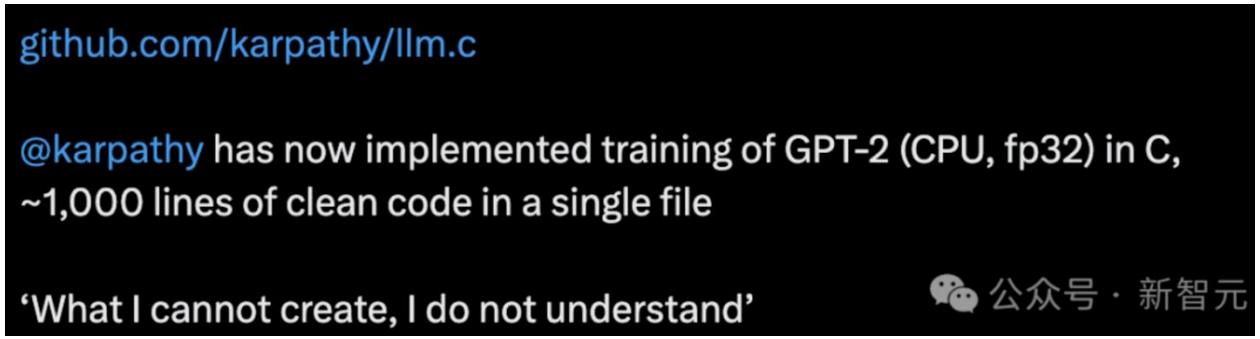
我们为什么要学习程序设计？

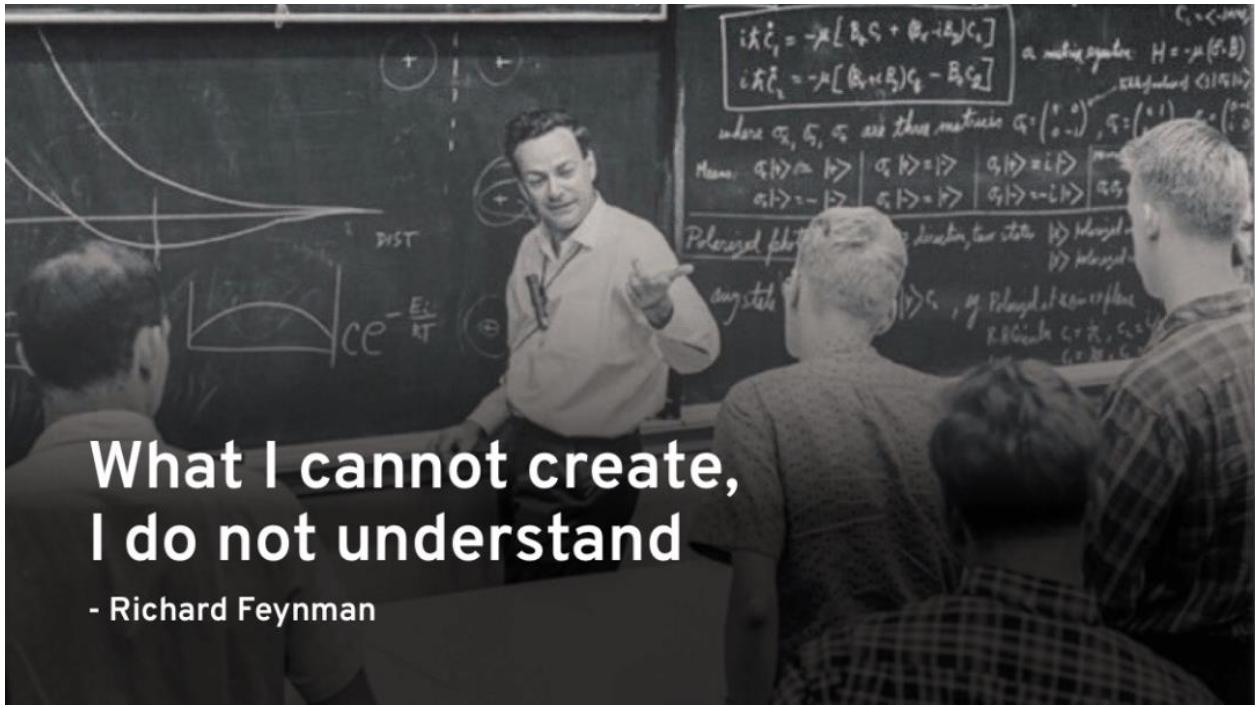
Why would you need to learn programming?

为什么要学程序设计

- 机器战胜了人类冠军并没有导致围棋这项运动的消亡

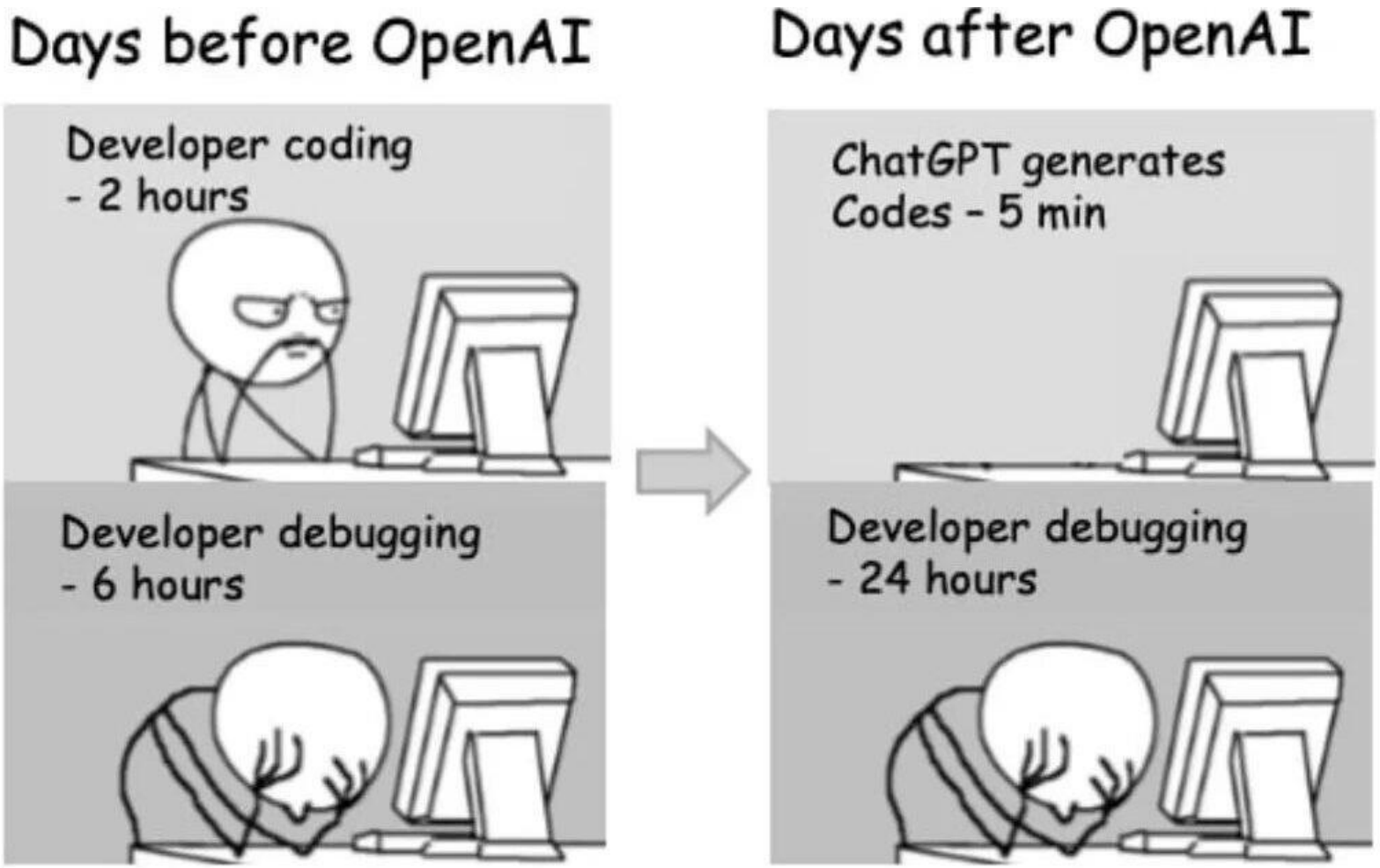
为什么要学程序设计 ：“理解是创造的基石”





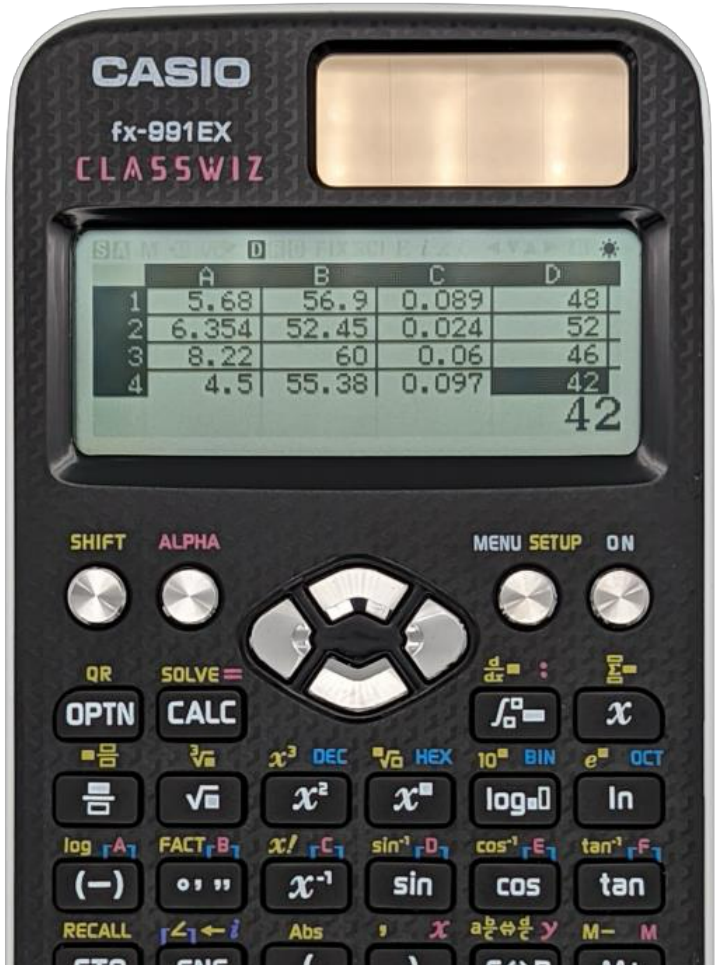
- 理解是创造的基石

为什么要学程序设计 ：“理解是创造的基石”



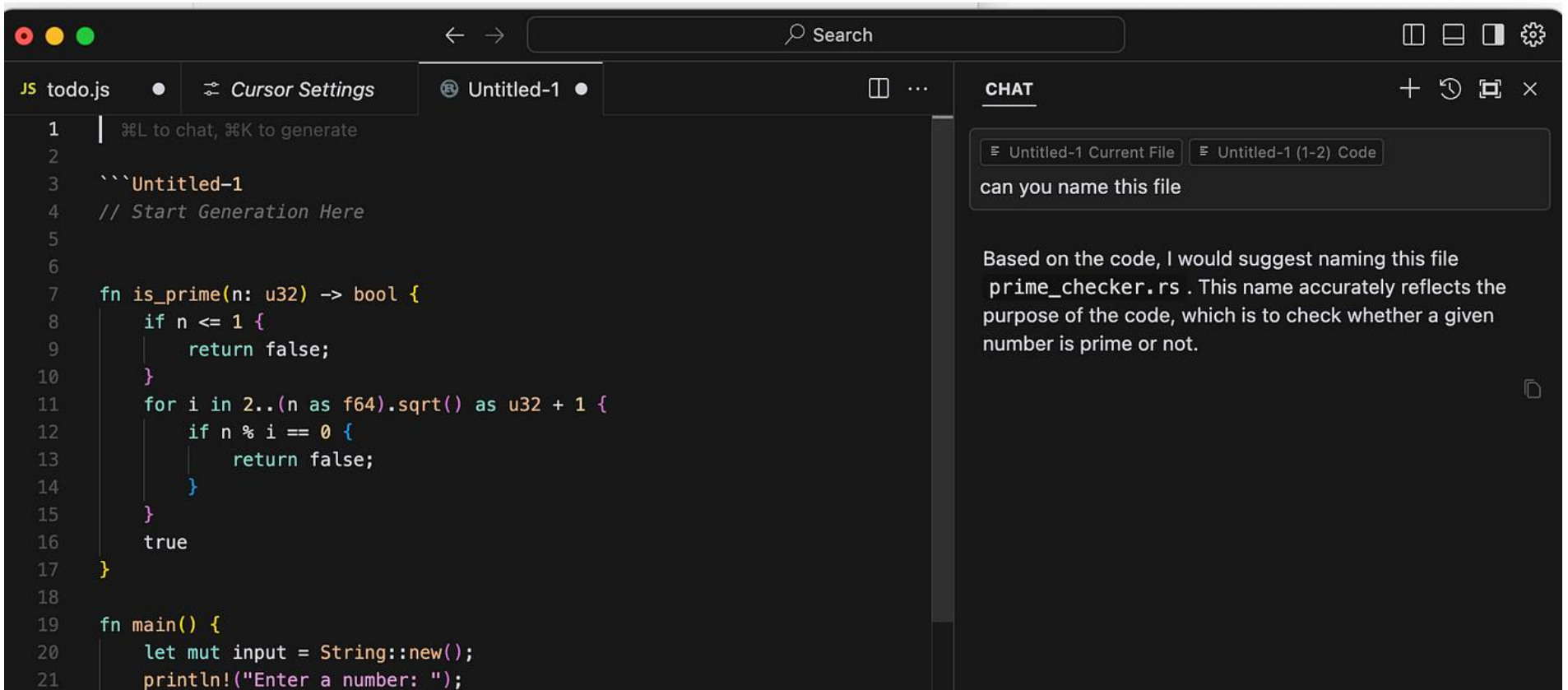
为什么要学程序设计：抓住核心

- 正如计算器的应用无法代替基础数学课。课程中所涵盖的逻辑是核心



为什么要学程序设计

- 正如计算器的应用无法代替基础数学课。课程中所涵盖的逻辑是核心
- 对于代码的校验和审核，需要扎实的程序设计基础知识。
  - 如何选取排序算法？冒泡排序 vs 归并排序？



那，为什么要学C++

- C++衍生出多种程序语言，很多程序设计的“源头”可以在C++的设计思路中找到答案
- 在AI时代，C++依然有不可替代的作用，例如机器人、智能硬件

那，为什么要学C++

- C++衍生出多种程序语言，很多程序设计的“源头”可以在C++的设计思路中找到答案
- 在AI时代，C++依然有不可替代的作用，例如机器人、智能硬件

为什么是“程序设计思想与方法”而不是“初级C++程序设计”

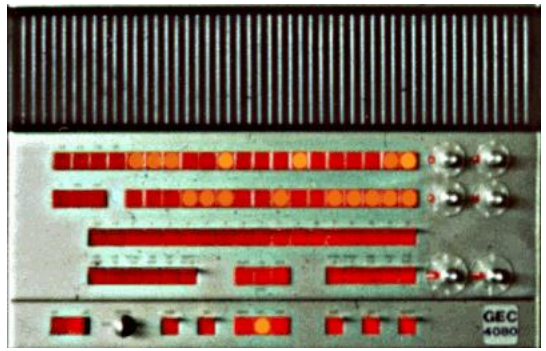
- 什么是Computer Science（CS）

- 程序设计就是**教**计算机如何解决问题。

- 我们如何教会计算机解决问题？
  - 了解计算机有哪些基本功能
  - 熟悉一门与计算机进行交流的语言
  - 掌握设计的思路与方法

计算机的组成

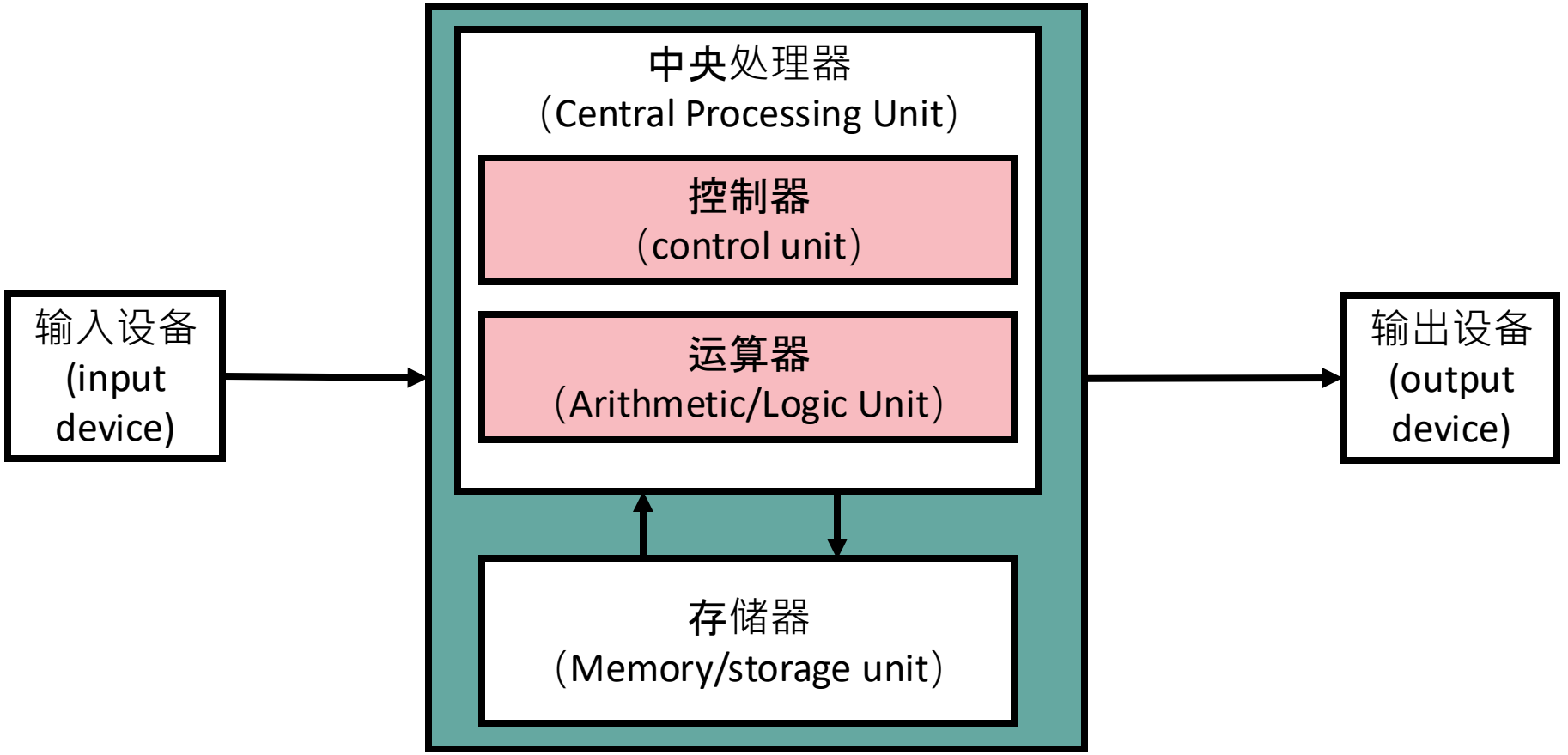
- 计算机，也被称之为“电脑”，是一种能够按照事先存储的程序自动、高效地对数据进行**输入**、**处理**、**存储**和**输出**的系统
- 数据：描述事物的符号记录
- 计算机组成：
  - 硬件：计算机的“躯壳”
  - 软件：计算机的“灵魂”



1970年代的“小型机”

计算机的组成

- 五大组成部分：冯·诺伊曼架构 （The von Neumann architecture）



计算机的组成

- 五大组成部分：冯·诺伊曼架构 （The von Neumann architecture）

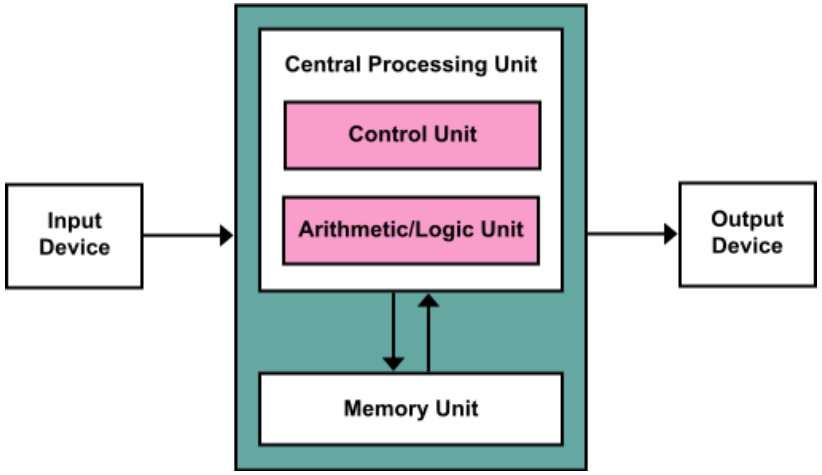
1.五个部分组成

- 输入
- 输出

} I/O

- 存储器
- 计算逻辑单元
- 控制单元

} CPU

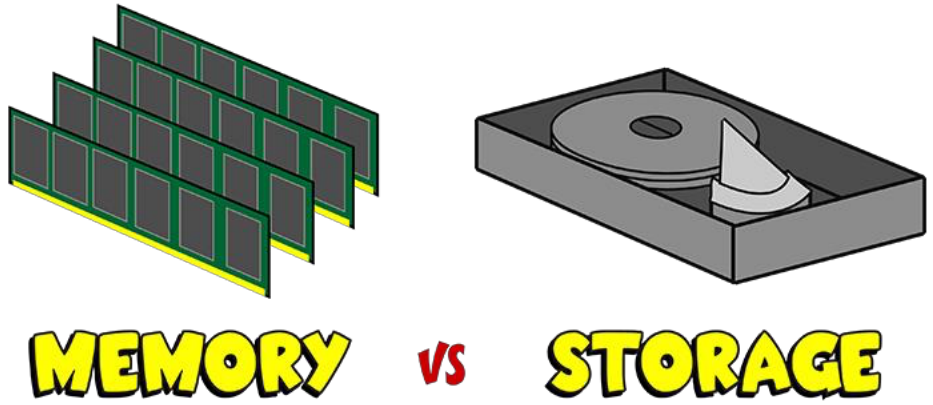


2. 三个关键概念:

- 指令和数据存放在**同一个可读写**的存储器之中
- 存储器单元**可以被寻址**，而不管里面存放什么类型的数据
- 程序总是**自上而下的串行执行**

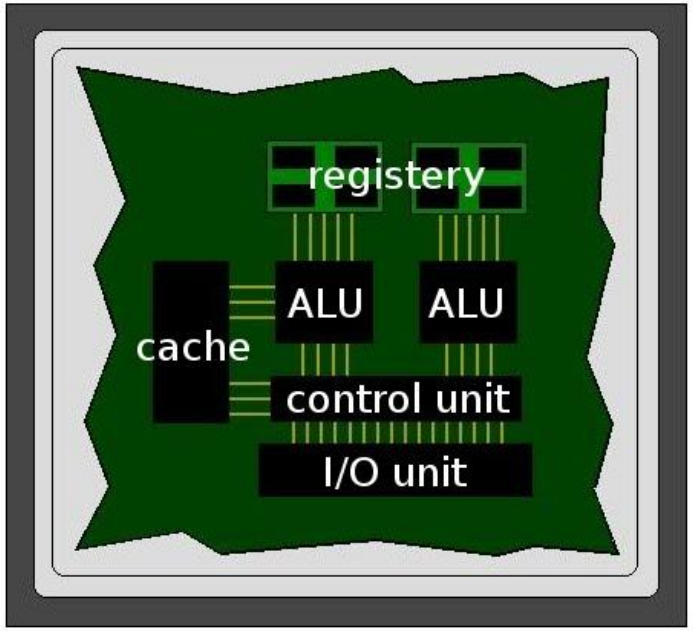
存储器

- 内存储器（Memory）和外存储器（Storage）
- 内存
  - 保存正在运行的程序代码和数据
  - 内存的最小单元是**比特（bit）**，一个bit存储一个二进制位。一般8个bit组成一个**字节（byte）**，若干个byte组成一个**字（word）**
  - 在一般的机器中，内存按字节编址，内存大小也是按字节计量
  - 关机后，内存的数据全部丢失
- 外存储器
  - 存储量大、成本低
  - 不能直接与运算器、控制器交换信息



CPU

- CPU由两个部件组成：
  - 控制器(Control unit, CU)
  - 运算器(Arithmetic-Logic Unit, ALU)



CPU-控制器

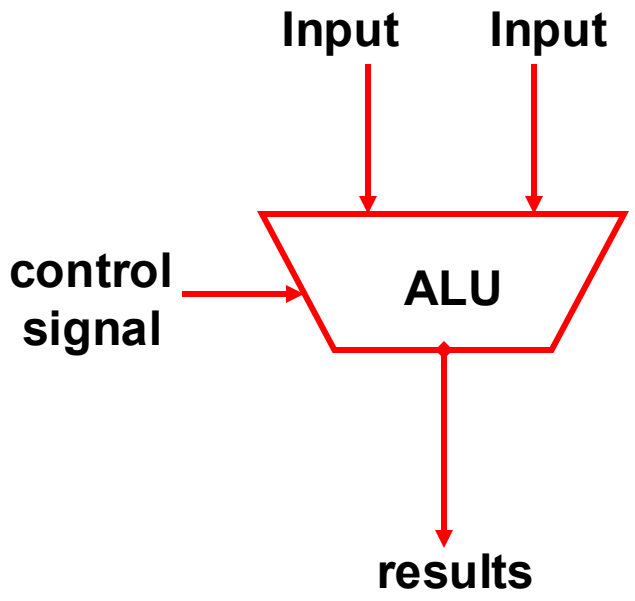
- 控制器控制计算机的其余部分如何完成程序的指令
  - 指挥CPU和输入输出设备之间的控制信息的传送

如何实现的？

- 控制器中有两种寄存器(register)：
  - 指令寄存器(Instruction register，IR)：保存当前正在执行的指令
  - 程序计数器 (Program counter，PC)：保存下一条要执行的指令地址
- 控制器工作过程：
  - 取下一条指令：按程序计数器（PC）指定的地址到内存中取出下一条指令，存入指令寄存器（IR）。
  - 解码指令：将指令解码成一系列的控制信号
  - 执行指令：将控制信号发送给相关部件，执行相应的运算

CPU-运算器

- 由一个多功能运算器和一组数据寄存器组成
- 可以执行算术运算和逻辑运算
  - 具体执行何种运算由控制器发送的控制信号决定
  - 有两个输入
  - 输出的结果保存在寄存器中



输入（Input）和输出（Output）设备

- 输入设备将人能理解的符号转换成计算机能处理的符号。
  常用的输入设备有：键盘、鼠标等。
- 输出设备将计算机的输出转换成人能理解的输出。常用的
  输出设备有：显示器、打印机、音响设备等。

Outline

- 课程介绍
- 什么是计算机
  - 计算机的组成
- 什么是程序
- 程序设计流程概览

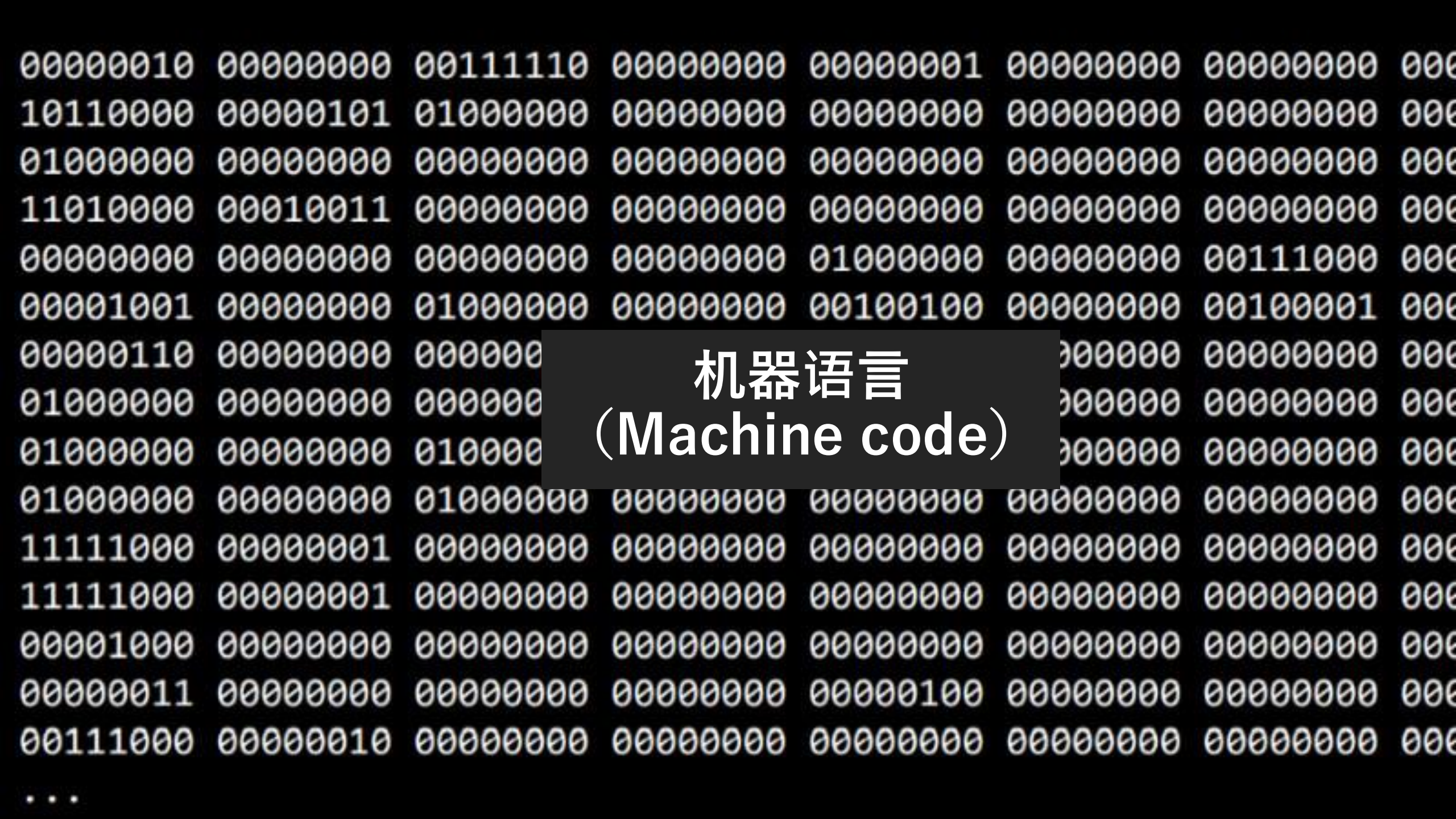
计算机程序（软件）

- 软件决定了计算机能做什么
- 软件可以分为系统软件和应用软件。
  - 系统软件居于计算机系统中最靠硬件的部分，它将计算机的用户与硬件隔离。系统软件与具体的应用无关，但其他的软件要通过系统软件才能发挥作用。
    - 常用系统软件：操作系统、编译器等。
  - 应用软件是为了支持某一应用而开发的软件。如游戏、文本编辑器等。
- 创造一个程序的过程就叫做编程

软硬件协同的例子:
iPhone (硬件) + iOS (操作系统) + App Store (第三方应用)

编程语言

- 程序设计语言是：人与计算机进行交流的一种语言。
- 为什么不用自然语言与计算机交互？
  - 自然语言存在严重的**二义性**
  - 很难有精确的语法（syntax）和语义（semantics）
- 因此，人们发明了不同层次的程序设计语言：
  - 机器语言（machine code）
  - 汇编语言（Assembly language）
  - 高级语言（High-level language）


00000010 00000000 00111110 00000000 00000001 00000000 00000000 000
10110000 00000101 01000000 00000000 00000000 00000000 00000000 000
01000000 00000000 00000000 00000000 00000000 00000000 00000000 000
11010000 00010011 00000000 00000000 00000000 00000000 00000000 000
00000000 00000000 00000000 00000000 01000000 00000000 00111000 000
00001001 00000000 01000000 00000000 00100100 00000000 00100001 000
00000110 00000000 000000
01000000 00000000 000000
01000000 00000000 010000
01000000 00000000 01000000 00000000 00000000 00000000 00000000 000
11111000 00000001 00000000 00000000 00000000 00000000 00000000 000
11111000 00000001 00000000 00000000 00000000 00000000 00000000 000
00001000 00000000 00000000 00000000 00000000 00000000 00000000 000
00000011 00000000 00000000 00000000 00000100 00000000 00000000 000
00111000 00000010 00000000 00000000 00000000 00000000 00000000 000
...
机器语言
（Machine code）

什么是机器语言

- 每个语句用一组二进制数来表示：
  - 例如： 0000010000000001.是Intel8086 能理解的一条指令。功能为：将寄存器的值加“1”
- 因此，用机器语言写程序是**非常困难**的，读机器语言写的程序也是异常困难的
- 每种计算机都有自己的机器语言，这与计算机硬件设计有关

汇编语言

```
                ********************************************
                * FUNCTION: INITA - Initialize ACIA
                * INPUT: none
                * OUTPUT: none
                * CALLS: none
                * DESTROYS: acc A

0013            RESETA  EQU     %00010011
0011            CTLREG  EQU     %00010001

C003 86 13      INITA   LDA A   #RESETA   RESET ACIA
C005 B7 80 04           STA A   ACIA
C008 86 11              LDA A   #CTLREG   SET 8 BITS AND 2 S
C00A B7 80 04           STA A   ACIA

C00D 7E C0 F1           JMP     SIGNON    GO TO START OF MON
```

什么是汇编语言

- 使用缩写（abbreviation）和助记符（mnemonic）来代替机器语言的0和1的比特串
- 汇编程序：将汇编语言写的程序翻译成机器语言的程序
  - 之前的加法运算例子即为: ADD AL, 1
  - 需要汇编程序处理后机器才懂

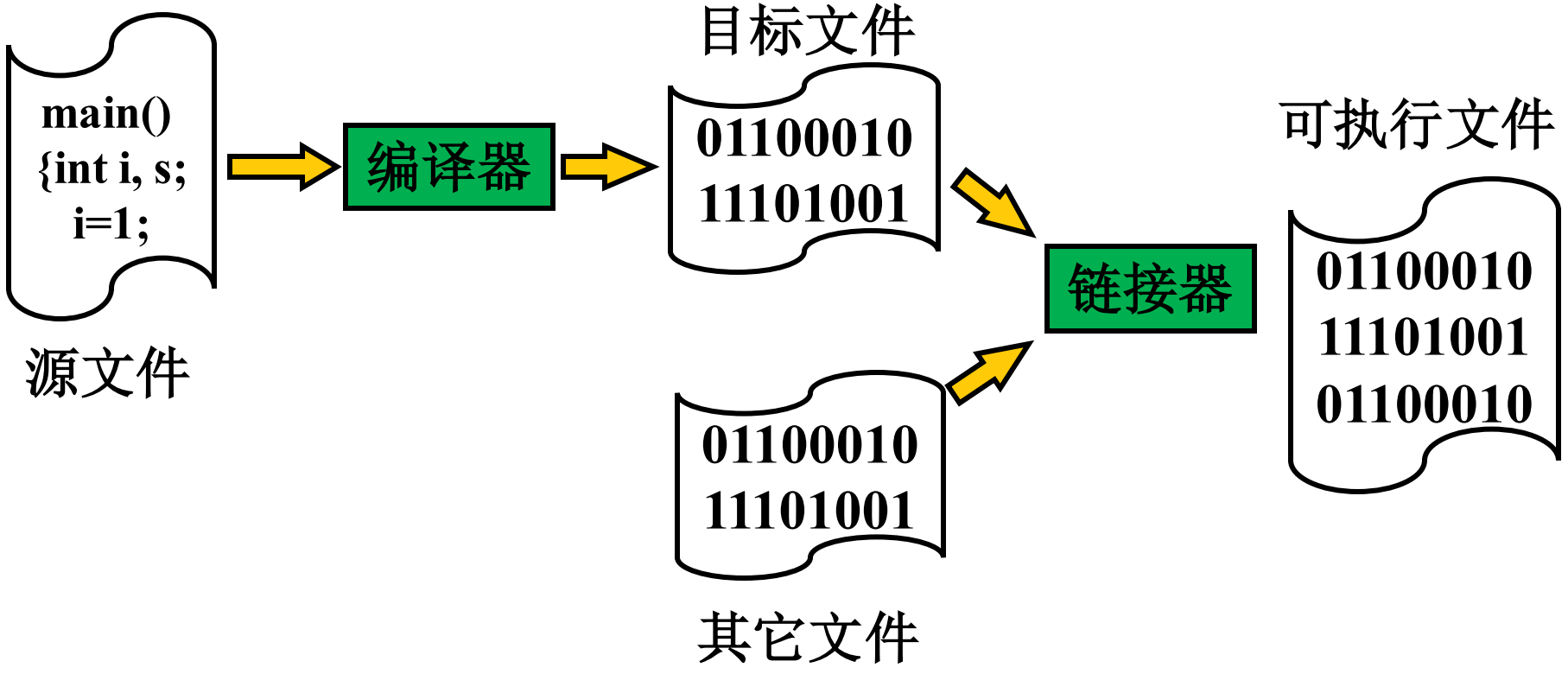
- 部分解决了机器语言的可读性问题，但没有解决功能简单的问题以及可移植性问题（在不同操作系统和不同的硬件配置下源程序很可能不同）

高级语言

什么是高级语言

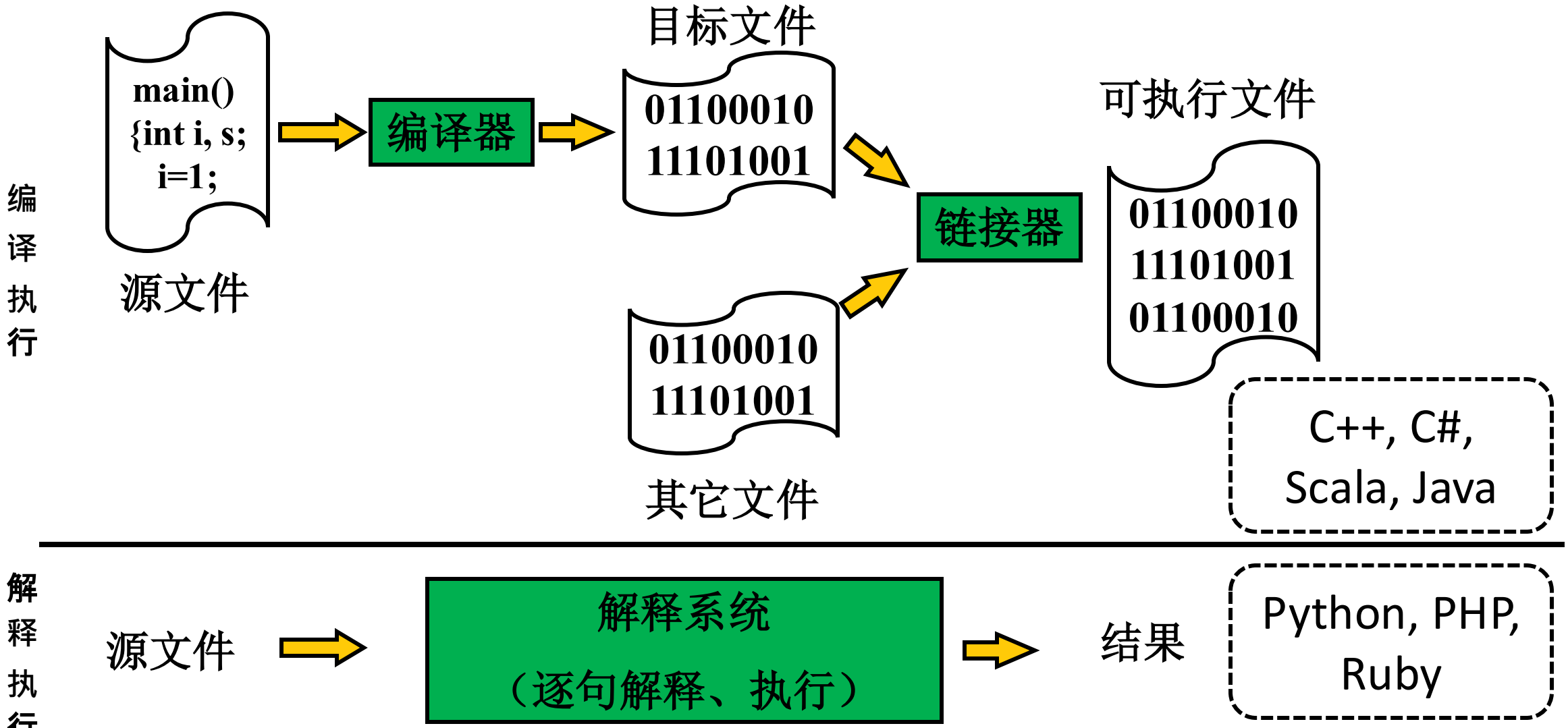
- 类似于英语的语言，适合于人理解
  - 如: x = y + 1.
- 需要编译器(compiler)或解释器 (Interpreter) 翻译后机器才懂.
- 编译器：将过程化语言写的程序（源代码）翻译成机器语言的程序（目标代码）
- 解释器：逐句解释源程序并执行，不保存目标代码。

编译执行 vs 解释执行


main()
{int i, s;
i=1;
源文件
编译器
目标文件
01100010
11101001
01100010
11101001
其它文件
链接器
可执行文件
01100010
11101001
01100010


源文件
解释系统
（逐句解释、执行）
结果

编译执行 vs 解释执行



[扩展阅读] Python是如何工作的：https://towardsdatascience.com/how-does-python-work-6f21fd197888

https://blog.csdn.net/tjxy_20081/article/details/8937687?utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-1.no_search_link&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7Edefault-1.no_search_link

高级语言的特点

- 具备了一定的机器独立性，使用户可以专注于**解决问题**的方法。但某些方面还是受到机器的限制。
- 为了解决移植性问题，ANSI制订了一系列的标准。
- 高级程序设计语言有很多种，2019年编程语言排行榜前10个语言是：Java，C，Python，C++，Visual Basic .NET，JavaScript，C#，PHP，SQL，Objective-C。

https://spectrum.ieee.org/top-programming-languages/#/index/2021/0/0/1/0/1/30/1/40/1/100/1/100/1/75/1/15/1/15/1/15/1/50/1/25/1/50/

Outline

- 课程介绍
- 什么是计算机
  - 计算机的组成
- 什么是程序
- 程序设计流程概览

程序设计过程

基本步骤：算法设计 ➔ 编程 ➔ 编译与链接 ➔ 调试

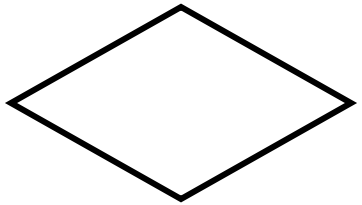
- 算法：使用计算机来解决某个问题的方案
- 难点：计算机基本功能简单 VS 任务非常复杂
- 算法特点
  1. 表述**清楚、明确、无二义性**
  2. 有效性，即每一个步骤都切实可行
  3. 有限性，即可在有限步骤后得到结果
- 算法表示
  - 自然语言
  - 流程图
  - 伪代码

流程图

求两个数相除算法

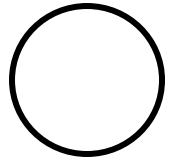
开始或结束

输入输出



选择

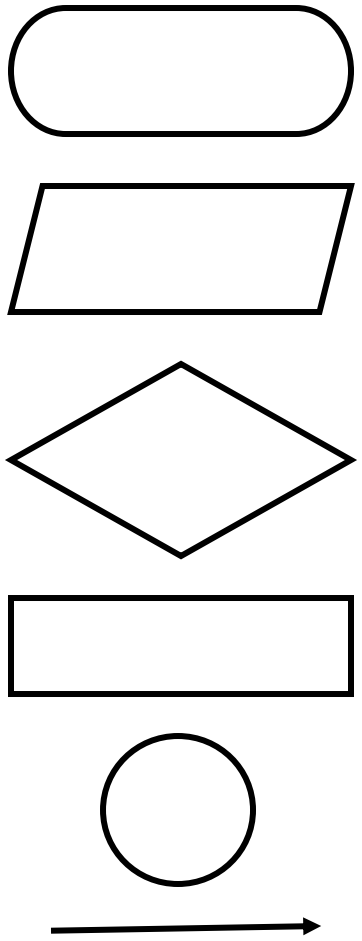
处理



链接

流程线

流程图



开始或结束

输入输出

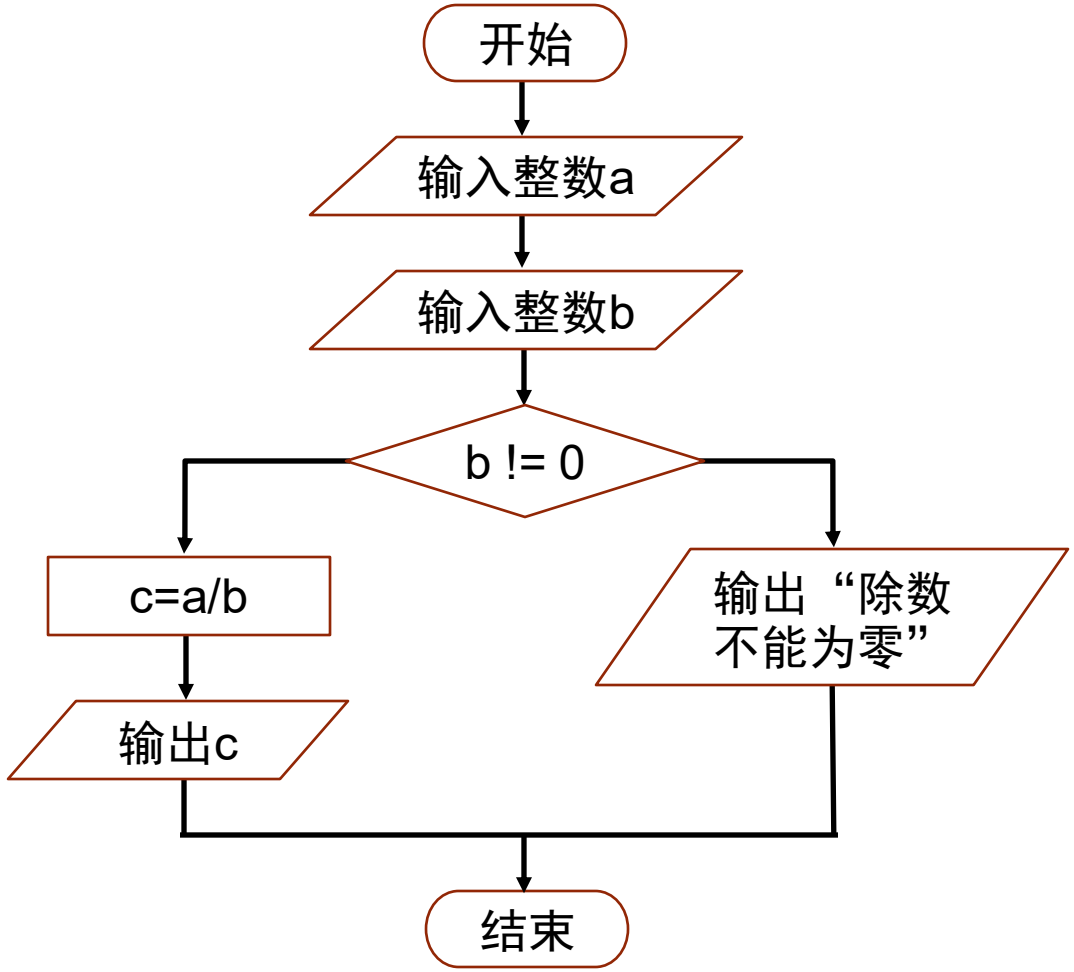
选择

处理

链接

流程线

求两个数相除算法



准备上机课，上机地点：电院4号楼，204+202

- 周五18:00
- 带好自己的电脑（充好电）