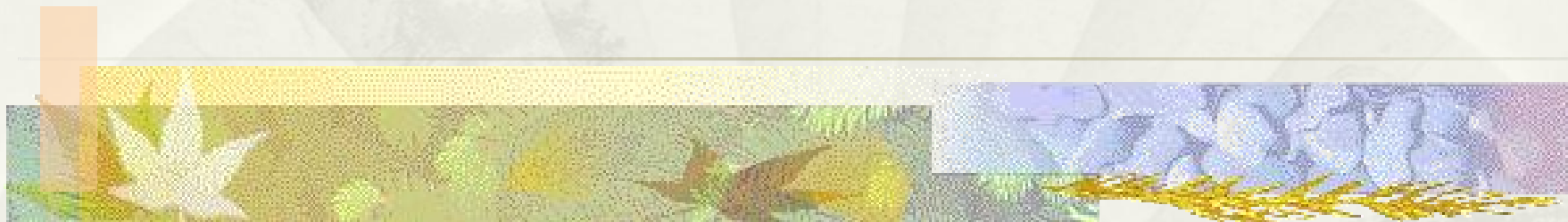


规格严格 功夫到家



第1章 绪论



哈尔滨工业大学（深圳）
计算机科学与技术学院

课件.版权：哈尔滨工业大学.苏小红 sxh@hit.edu.cn



基本信息

计算机科学与技术学院

授课教师：刘洋

助教：刘洋麟 田宇琛 俞陈佳 张玉玺

<http://yangliu.info>

办公室：L1703

课程群号：744928573

主要教材：《C语言程序设计（第三版）---苏小红 等著》

成绩评定：作业20分+实验30分+期末50分



高级语言程序设计...

群号：744928573



扫一扫二维码，加入群聊。



课程教学目标

- 掌握用计算机求解问题的基本思维方法
- 掌握高级语言程序设计的基本知识
- 掌握计算机软件设计与实现的基本方法
- 具备用计算机解决实际问题的能力和基本的程序调试能力



**How to Think
Like a Computer Scientist**



**How to program
Like a Computer Scientist**

第1章 学习内容

- 计算机和程序设计的基础知识
- C语言简介

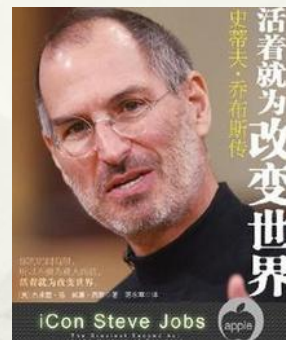


问题1：你了解计算机吗？

- 为什么计算机也被俗称为“电脑”？
 - * 期望计算机有朝一日能像人脑一样聪明
- 让机器能像人一样思考，一直是人类的梦想
- 很多科幻故事描绘了计算机真正成为电脑时的景象
 - * 《变形金刚》里的机器人大战
 - * 《AI（人工智能）》里探讨的人类与机器的感情关系
 - * 《终结者》里机器人企图消灭人类
 - * 最有想象力的是《Matrix（黑客帝国）》

问题2：你了解计算机领域的哪些人？

- 你了解计算机科学领域中的哪些著名的人物？



- 阿兰·图灵 (Alan Turing)

- * 现代计算机科学之父和人工智能之父的英国数学家
- * 1950年发表题为“Computing Machinery and Intelligence”的论文，首次提出“机器思维”的概念，并从“机器能否思考”这一问题出发，提出一个判断机器是否有智能的测试标准——“图灵测试” (Turing Testing)



问题3：什么是“图灵测试”？

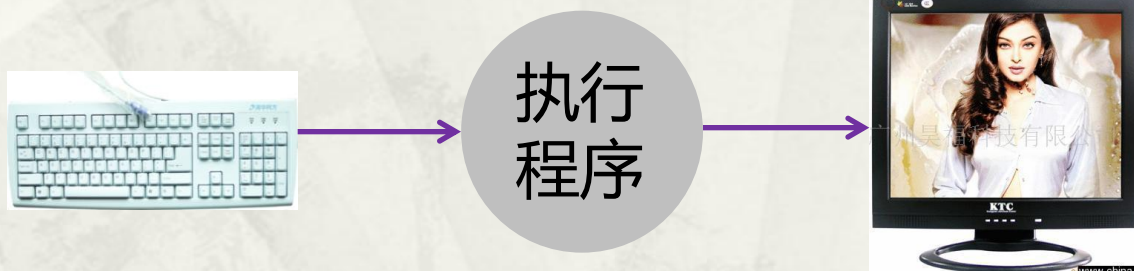
- 图灵提出的一种判断机器是否具有智能的标准（方法）
 - * 测试人在不接触被测试者（一个人和一台机器）的情况下，通过一些终端设备（如键盘）向被测试者提问
 - * 如何判断这台机器通过了测试，并具有与人类相当的智能呢？

更换测试人或被
测试者中的人？



问题4：什么是计算机系统？

- 一种能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



- 计算机系统（Computer System） = 硬件 + 软件



问题5：什么是硬件？

■ 硬件(Hardware)

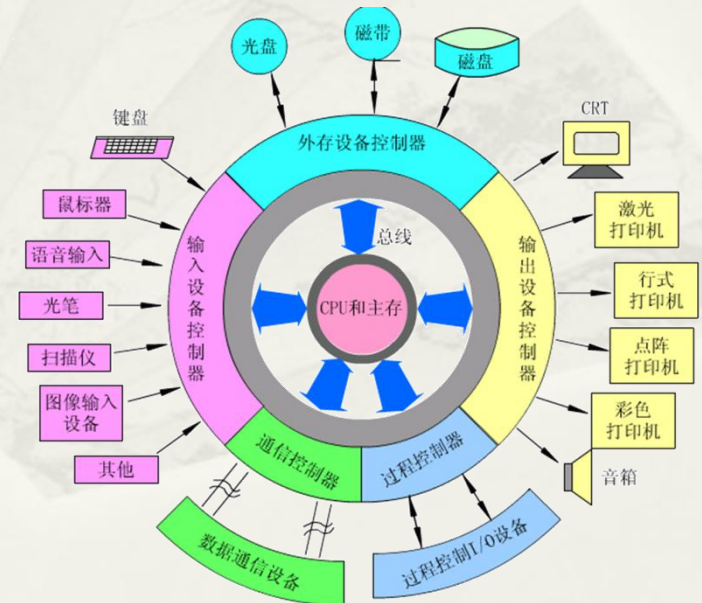
- * 计算机系统中所有实体部件和设备的统称
- * 由中央处理器(CPU)、存储器和输入/输出设备等核心部件组成
- * 在软件的配合下完成输入、处理、存储和输出等基本操作
- * 计算机是用来延伸人的能力的工具



眼睛和耳朵

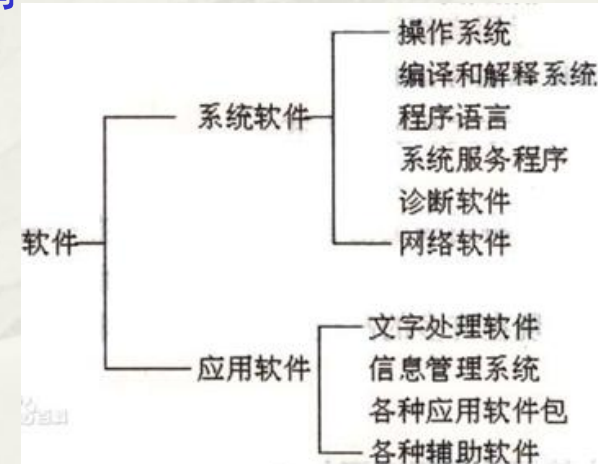


大脑



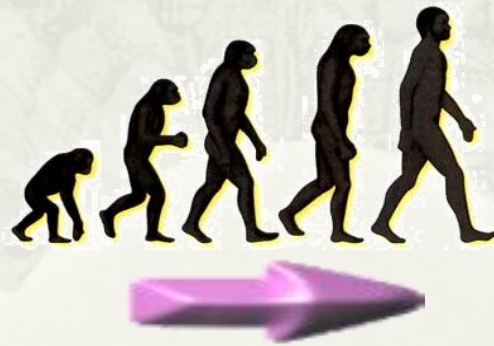
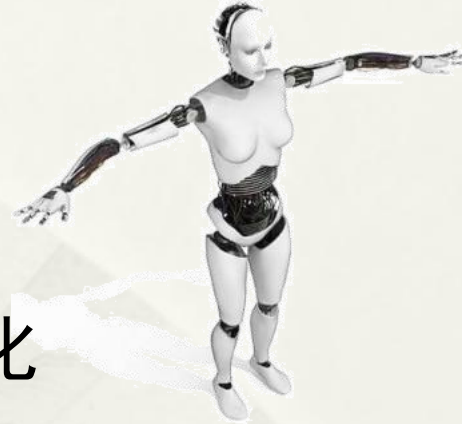
问题6：什么是软件？

- 人驾驭计算机的手段之一是编写软件（OS, Office, IE,...）
- 软件(Software)
 - * 系统软件—为使用计算机提供最基本的功能
 - * 操作系统—用户与计算机硬件系统之间的接口，管理计算机系统的各种软、硬件资源，使得它们可以协调工作，最内层
 - * 支撑软件—环境数据库、接口软件和工具软件
 - * 应用软件—直接面向特定应用的软件
 - * 图像处理，文字处理，计算机动画，CAD...



问题7：软件与硬件的关系是什么？

- 硬件——计算机的躯壳和肉体
- 软件——计算机的灵魂和思想
- 软件促进了计算机从裸机到计算机系统的进化



问题8：程序和软件有区别吗？

- 程序（Program）
 - * 为了实现特定目标或解决具体问题而用计算机语言编写的指令的有序集合
- 程序就是软件的全部吗？
- 程序、数据、文档之间的关系



厨师	软件开发人员
烹饪方法	软件中的程序
菜谱	软件的文档
食材	软件中的数据
整个烹饪过程	软件的实现过程



问题9：计算机能读懂什么语言？

- 计算机只能识别二进制指令
- 机器语言（Machine Language）
 - * 用机器语言编写的一组机器指令的集合，称为机器代码（Machine Code），简称机器码
 - * 用一系列二进制0和1组成的二进制代码代表不同的机器指令
 - * 机器语言编写的1+1程序——犹如天书！



```
10111000
00000001
00000000
00000101
00000001
00000000
```

计算机的母语
是什么？



问题10：如何让程序员远离二进制？

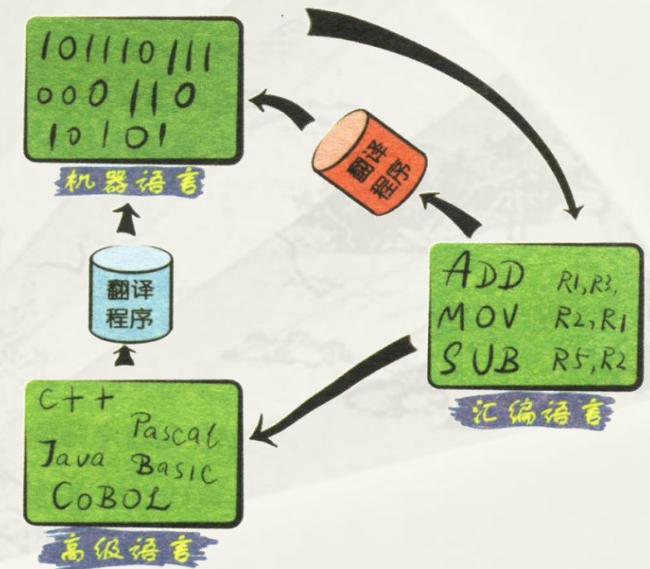
- Both learn a third-party language
- 用助记符代表机器指令的操作码
 - 用一个汇编器（Assembler）将这些符号翻译成机器可识别、可执行的二进制指令——汇编过程
- 汇编语言（Assembly Language）
 - 低级语言，与计算机硬件的距离的级别较低，对硬件依赖性强

MOV AX, 1	10111000 00000001 00000000
<hr/>	
ADD AX, 1	00000101 00000001 00000000



问题11：如何实现更有效的人机交流？

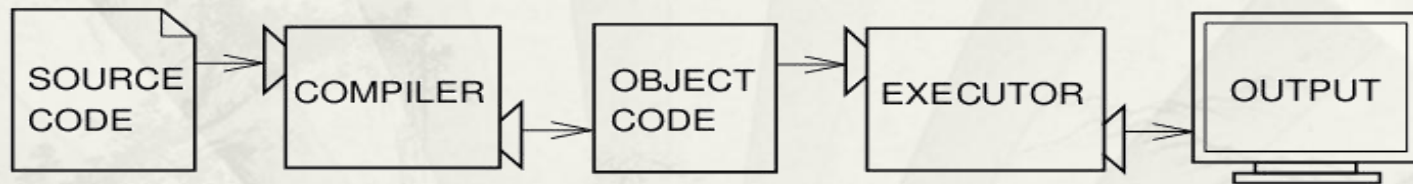
- 设计更贴近人类自然语言和数学语言的计算机语言
- 高级语言（ High-level Languages ）
 - “高级” 在哪里？
 - 可以在一个更高的级别上进行编程
 - 不像低级语言那样依赖于硬件
 - 更远离机器语言
 - 更易学易用，具有可移植性



问题12：如何将高级语言翻译成机器语言？

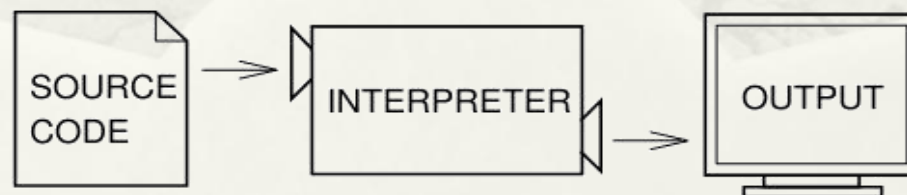
■ 编译型语言

- * 通过编译器（Compiler）把源程序全部翻译成逻辑上与之等价的机器语言表示的目标代码（Object Code），才能被计算机识别和执行



■ 解释型语言

- * 解释程序（Interpreter）边扫描边解释，语句被逐条地解释成可执行的机器指令，不产生目标程序——“同声传译”
- * 有些解释语言（如Java和C#）采用了“先编译、后解释”的方法来提速



高级语言的特点

- BASIC语言编写的1+1程序
- C语言编写的1+1程序

```
PRINT 1+1
```

```
#include <stdio.h>
int main()
{
    printf("%d\n", 1+1);
    return 0;
}
```

- 既非英语，也非机器语言，但更像英语
- 高级语言程序必须通过编译器或解释器翻译成机器语言形式的目标代码，才能被计算机识别和执行

初露锋芒



百花齐放



开宗立派



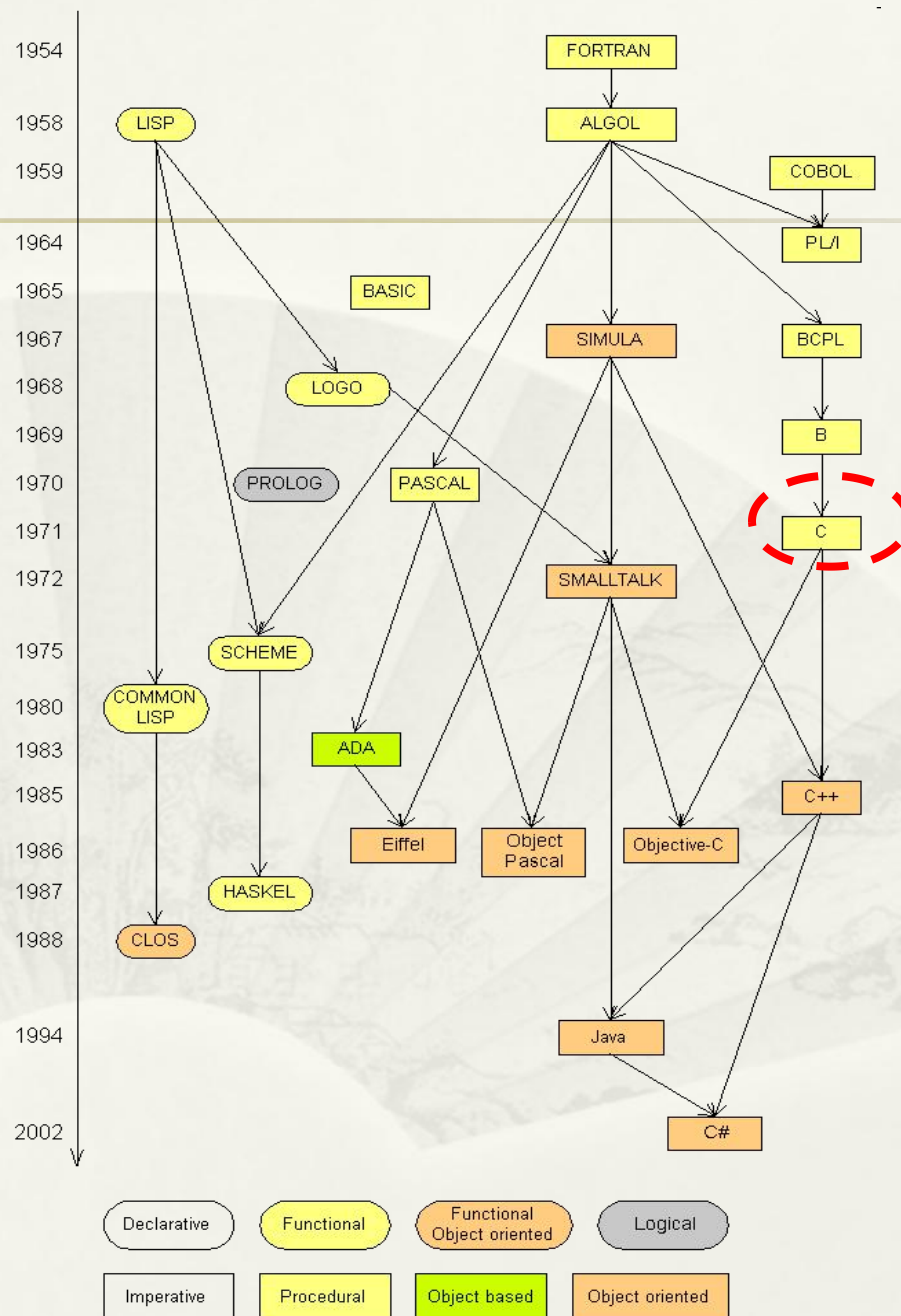
内外兼修



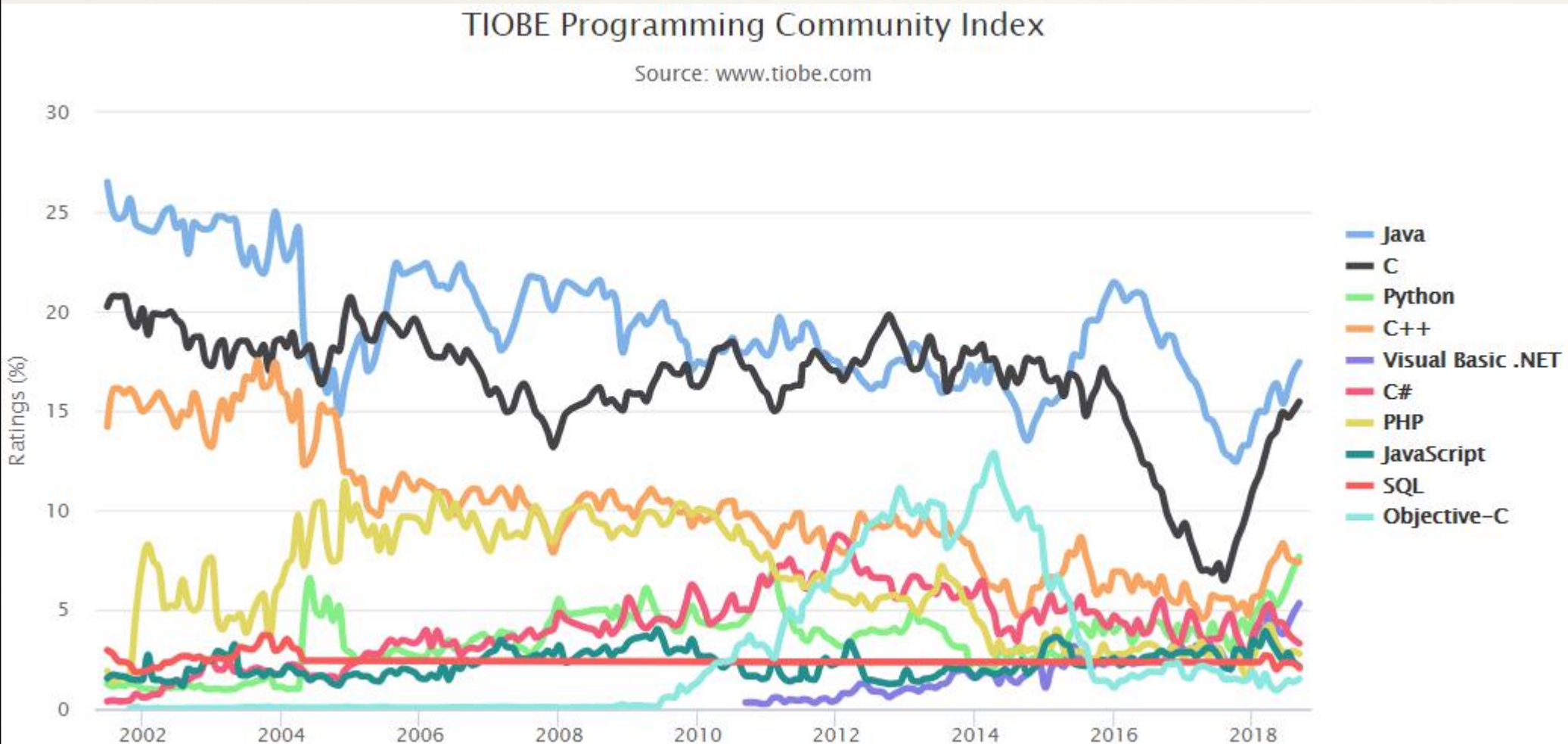
互联网时代

高级语言的发展历程

- 1954年约翰.巴克斯发明第一个高级语言FORTRAN
- 百家争鸣，据不完全统计2500种
 - <http://people.ku.edu/~nkinn ers/LangList/Extras/langlist .htm>可看到其列表和简介
- 影响最大、寿命最长的非C语言莫属



编程语言受欢迎程度排行



<http://www.tiobe.com/tpci.htm>

高级语言的分类

语言	典型代表	特点
过程式语言 (Procedural)	C	命令驱动，面向动作（语句），即将计算看作是动作（语句）的序列，程序由一系列的语句组成
函数式语言 (Functional)	LISP	注重程序实现的功能，程序的编写过程是从已有函数出发构造出更复杂的函数，应用这些函数直至计算求得最终结果
面向对象语言 (Object-Oriented)	SmallTalk、C++、Java	程序的编写过程是直接从问题本身出发分析问题，并把数据和操作封装在一起构成对象，用数据描述对象属性，用对数据的操作描述对象行为
逻辑式语言 (Logical)	PROLOG	基于规则的具有推理功能的语言

C语言标准

- 1978年，Dennis M.Richie和Brian W.Kernighan合著影响深远的名著《The C Programming Language》
- 1982年，美国国家标准化协会（American National Standards Institute, ANSI）成立C标准委员会
- 1989年，发布第一个完整的C语言标准，简称C89
 - 习惯上也称为ANSI C（标准C）
 - 1990年被国际标准化组织ISO采纳，也称为C90
 - 1999年修正和完善之后，ISO发布了C99
 - 2011年修正和完善之后，ISO发布了C11

问题13：为什么要学C语言？

■ 是一种高级语言

- * 并不“高级”，只是相对低级语言在一个高的级别上进行编程
- * 实际上是一种介于高级语言和低级语言之间的语言，运行效率高
- * 透过现象看本质，透过C语言窥探计算机底层的工作原理

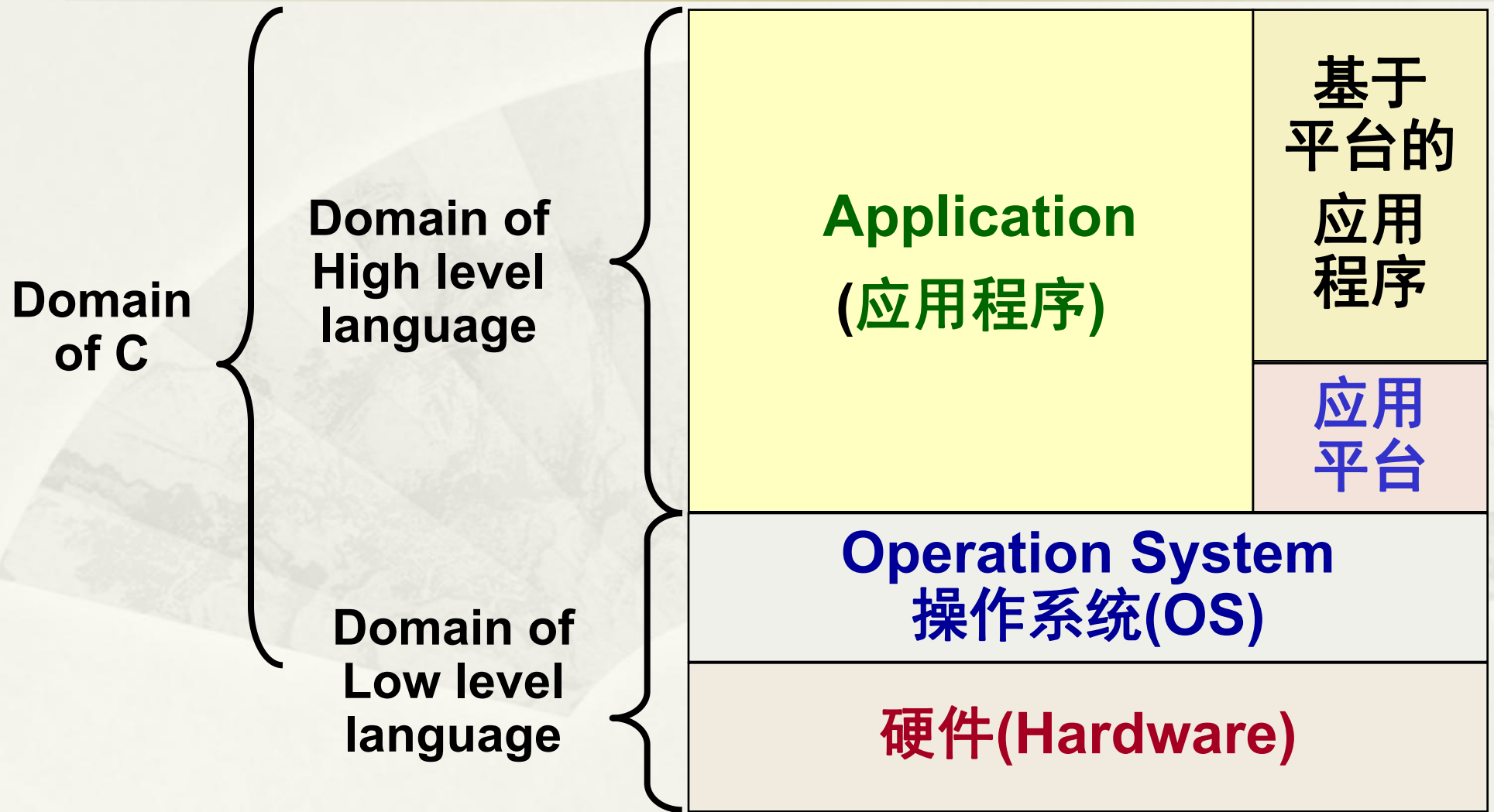
■ 很多流行语言、新生语言都借鉴了它的思想、语法

- * 从C++，到Java，再到C#
- * 学好C是学习这些流行语言的基础，了解程序设计的基本思想
- * 交流、笔试、面试时最常见的语言

问题13：为什么要学C语言？

- 历史悠久，战勋卓著
 - * 诞生于20世纪70年代初
 - * 成熟于80年代
 - * 很多重量级软件都是用C语言写的，维护已有的C代码
- 上天入地，无所不能
 - * 几乎没有不能用C语言实现的软件
 - * 没有不支持C语言的系统

问题14: C语言适合做什么?



问题15：选择语言的标准是什么？

- 适合的才是最好的
- C语言最适合哪些应用？
 - * 编写系统软件（例如OS）和工具软件
 - * 运行效率要求较高的系统
 - 设备驱动程序，嵌入式系统（Embedded systems）
（工业机器人、家用机器人、智能手机和智能家电等）
 - 高性能、实时系统（Real-time systems）



问题16：计算机、程序员、程序之间的关系？

- 程序员（Programmer）
 - * 编写程序源代码（Source Code）的人
- 计算机、程序员、程序之间的关系

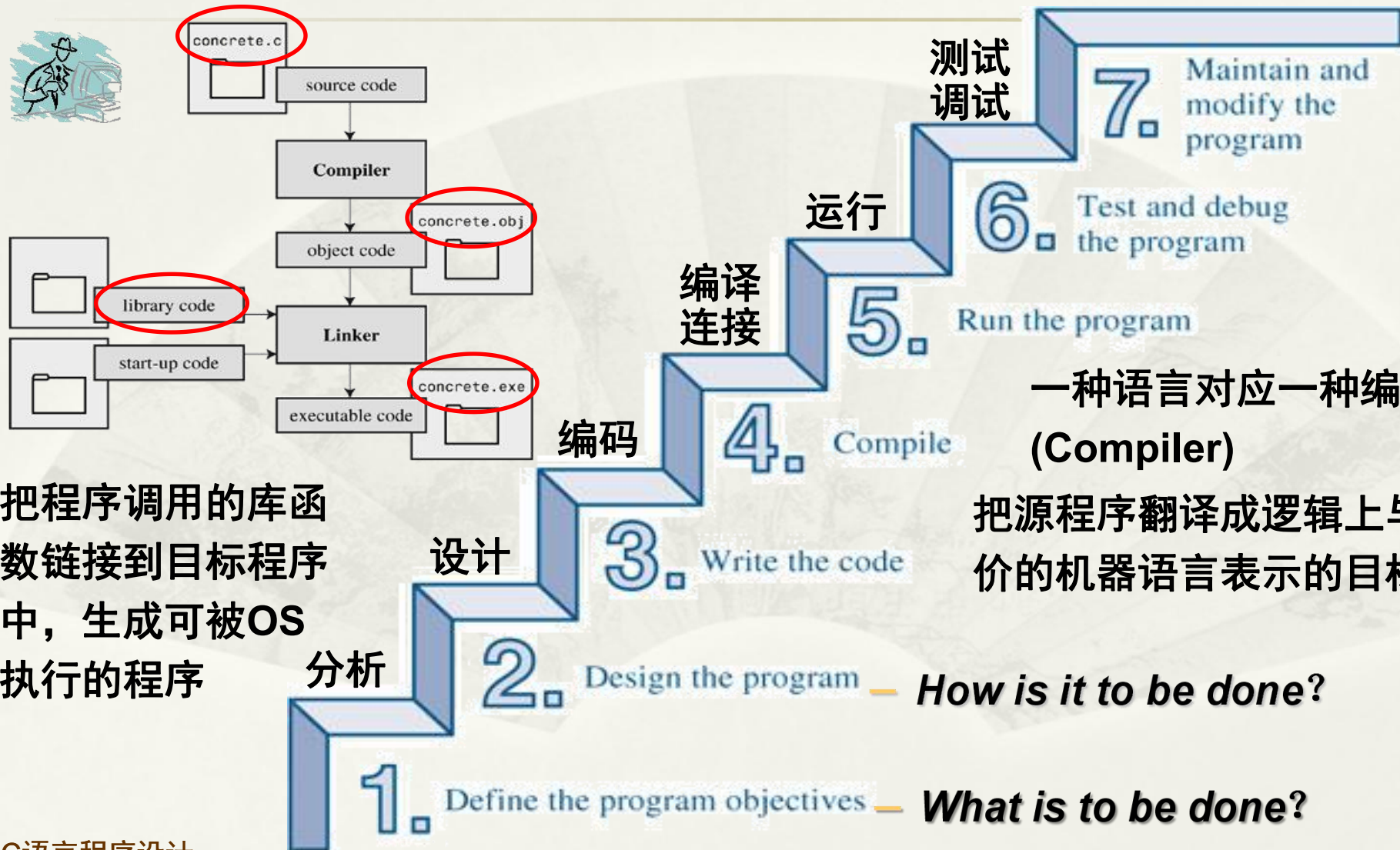


计算机	程序员	程序
乐队	指挥家	乐谱
军队	总司令	作战计划



程序的开发步骤

维护修改



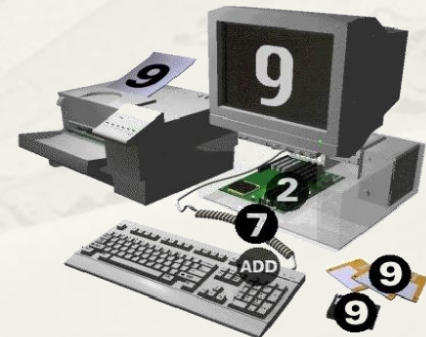
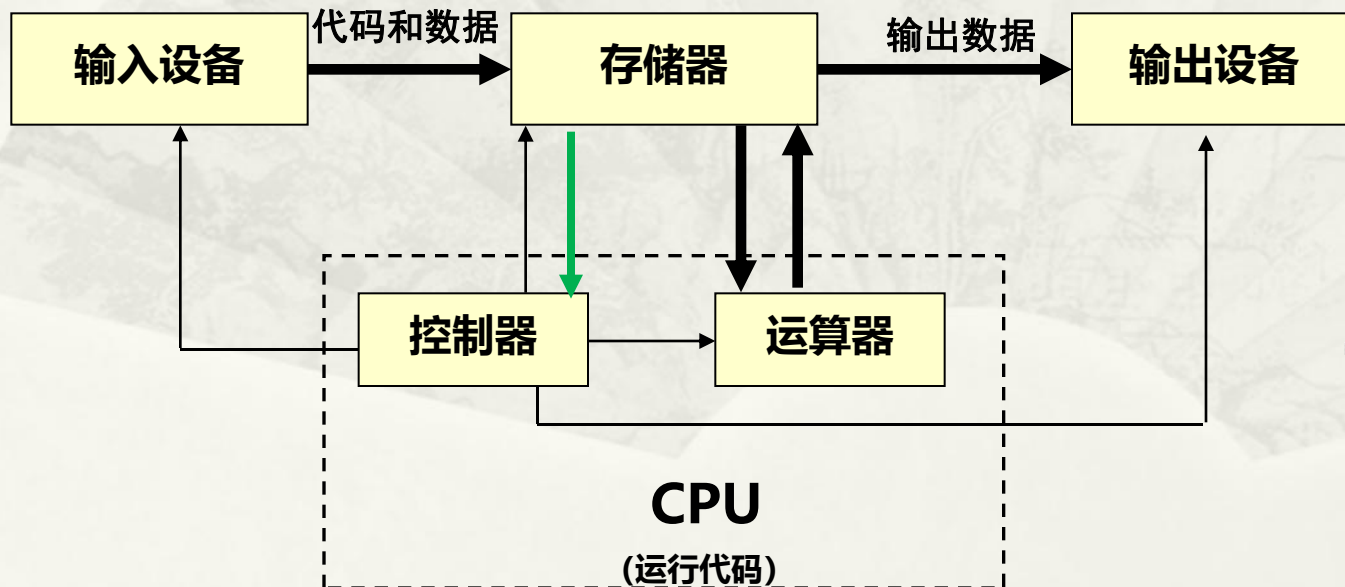
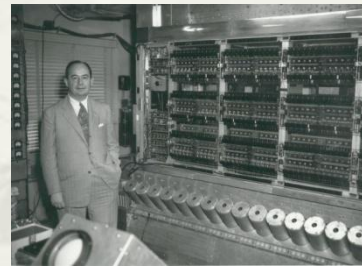
把程序调用的库函数链接到目标程序中，生成可被OS执行的程序

一种语言对应一种编译器 (Compiler)

把源程序翻译成逻辑上与之等价的机器语言表示的目标程序

问题17：程序在计算机内部是如何运行的？

- 冯·诺依曼（Von Neumann）率先在计算机中采用二进制，并且提出了著名的“冯·诺依曼机”体系结构
 - * 至今仍被几乎所有的计算机采用，鉴于冯·诺依曼在发明电子计算机中所起的关键性作用，被誉为“计算机之父”



问题18: C语言什么样?

简单的C程序: “Hello world!”

```
#include <stdio.h>
main()
{
    printf("Hello world!\n");
}
```

复杂点: 加法计算器

```
#include <stdio.h>

int Add(int a, int b)
{
    return (a + b);
}

main()
{
    int x, y, sum = 0;

    printf("Input two integers:");
    scanf("%d%d", &x, &y);
    sum = Add(x, y);
    printf("sum = %d\n", sum);
}
```

问题19：如何学好C语言？

万事开头难



百折不挠

师傅领进门，
出徒在个人
世界上没有从不练琴的钢琴家



练习、练习、再练习



持之以恒



作业

- 讨论题目一：C语言的过去现在和未来
 - 1. C语言的发展史
 - 2. C语言的现状和未来
 - 3. C语言的特点及优势
 - 4. C语言在本专业领域的应用
 - 5. 为什么要学C语言
- 讨论题目二（学期结束前）：深入了解一项人工智能相关的技术或项目

- http://www.iqiyi.com/w_19rru1dehh.html#curid=4197954809_6ddf83e499e7767d36923de9995d4371
- <https://www.bilibili.com/video/BV1xs411S77R?from=search&seid=13738257171074173913>