软件工程专业导论

复习

1. 以下关于计算机系统及其组成的叙述不正确的是（ ）?

- A. 鼠标和键盘属于计算机系统的输入设备
- B. 软件系统包括了系统软件、支撑软件和应用软件
- C.内存和硬盘都有存储功能，都属于计算机系统的外设
- D.计算机系统包括了硬件系统和软件系统

复习

1. 以下关于计算机系统及其组成的叙述不正确的是（ ）?

- [ ] A. 鼠标和键盘属于计算机系统的输入设备
- [ ] B. 软件系统包括了系统软件、支撑软件和应用软件
- [√] C.内存和硬盘都有存储功能，都属于计算机系统的外设
- [ ] D.计算机系统包括了硬件系统和软件系统

复习

2. 下列选项中，不属于面向对象编程语言的是（ ）?

- A. Visual 系列语言
- B. C语言
- C. C++
- D. Java

复习

2. 下列选项中，不属于面向对象编程语言的是（ ）?

- [ ] A. Visual 系列语言
- [√] B. C语言
- [ ] C. C++
- [ ] D. Java

复习

3. 软件工程方法发展历程的正确顺序是（ ）?

- A. 面向对象的方法->结构化方法->面向服务的SOA方法->构件化方法和Web Services->基于互联网与云计算的软件开发方法
- B.结构化方法->面向对象的方法->构件化方法和Web Services->面向服务的SOA方法->基于互联网与云计算的软件开发方法
- C.结构化方法->面向对象的方法->面向服务的SOA方法->构件化方法和Web Services->基于互联网与云计算的软件开发方法
- D.面向对象的方法->结构化方法->构件化方法和Web Services->面向服务的SOA方法->基于互联网与云计算的软件开发方法

复习

3. 软件工程方法发展历程的正确顺序是（ ）?

☐ A. 面向对象的方法->结构化方法->面向服务的SOA方法->构件化方法和Web Services->基于互联网与云计算的软件开发方法

☑ B.结构化方法->面向对象的方法->构件化方法和Web Services->面向服务的SOA方法->基于互联网与云计算的软件开发方法

☐ C.结构化方法->面向对象的方法->面向服务的SOA方法->构件化方法和Web Services->基于互联网与云计算的软件开发方法

☐ D.面向对象的方法->结构化方法->构件化方法和Web Services->面向服务的SOA方法->基于互联网与云计算的软件开发方法

复习

4. 已知A-Z的ASCII码是41H-5AH，请将下面一段ASCII码存储的文件解析出来，正确的是_____。

“0100 0111 0100 0101 0100 0111 0100 0110 0100 1000 0100 0010”

- A. GBHEGB
- B. HBFFEG
- C. GEGFHB
- D. HBGFGE

复习

4. 已知A-Z的ASCII码是41H-5AH，请将下面一段ASCII码存储的文件解析出来，正确的是_____。

“0100 0111 0100 0101 0100 0111 0100 0110 0100 1000 0100 0010”

- [ ] A. GBHEGB
- [ ] B. HBFFEG
- [√] C. GEGFHB
- [ ] D. HBGFGE

复习

5. 关于十进制245的下列说法不正确的是_____。

- A.它转换为二进制表示为1101 0101
- B.它转换为八进制表示为365
- C.它转换为十六进制表示为0F5

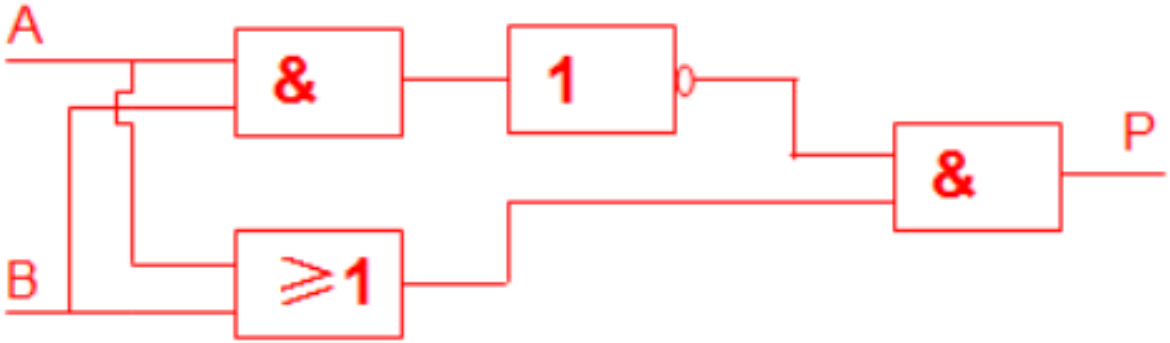
复习

5. 关于十进制245的下列说法不正确的是_____。

- [√] A.它转换为二进制表示为1101 0101
- [ ] B.它转换为八进制表示为365
- [ ] C.它转换为十六进制表示为0F5

复习

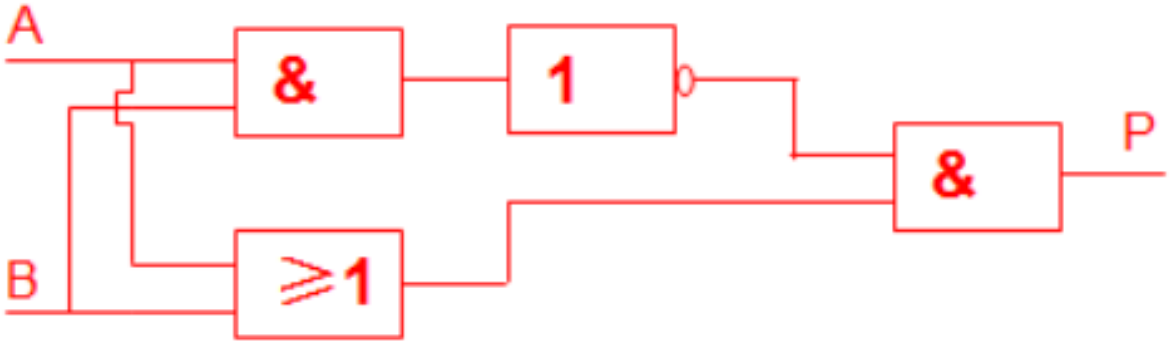
6. 该电路所实现的正确的逻辑运算为_____。



- A. P = A XOR B
- B. P = (A AND B) AND (A OR B)
- C. P = NOT (A AND B) AND (A AND B)
- D. P = (A OR B) AND (A AND (NOT B))

复习

6. 该电路所实现的正确的逻辑运算为_____。



- [√] A. P = A XOR B
- [ ] B. P = (A AND B) AND (A OR B)
- [ ] C. P = NOT (A AND B) AND (A AND B)
- [ ] D. P = (A OR B) AND (A AND (NOT B))

复习

7. 已知：关于 S_i 和 C_{i+1} 的逻辑运算式如下：

$S_i = ((A_i \text{ XOR } B_i) \text{ XOR } C_i$

$C_{i+1} = ((A_i \text{ AND } B_i) \text{ OR } ((A_i \text{ XOR } B_i) \text{ AND } C_i)$ ，问：

如果 $A_i=1$，$B_i=1$，$C_i=1$，则 S_i，C_{i+1} 的值为_____。

- A. 0，1
- B. 1，1
- C. 0，0
- D. 1，0

复习

7. 已知：关于 S_i 和 C_{i+1} 的逻辑运算式如下：

$S_i = ((A_i \text{ XOR } B_i) \text{ XOR } C_i$

$C_{i+1} = ((A_i \text{ AND } B_i) \text{ OR } ((A_i \text{ XOR } B_i) \text{ AND } C_i)$ ，问：

如果 $A_i=1$，$B_i=1$，$C_i=1$，则 S_i，C_{i+1} 的值为_____。

- [ ] A. 0，1
- [√] B. 1，1
- [ ] C. 0，0
- [ ] D. 1，0

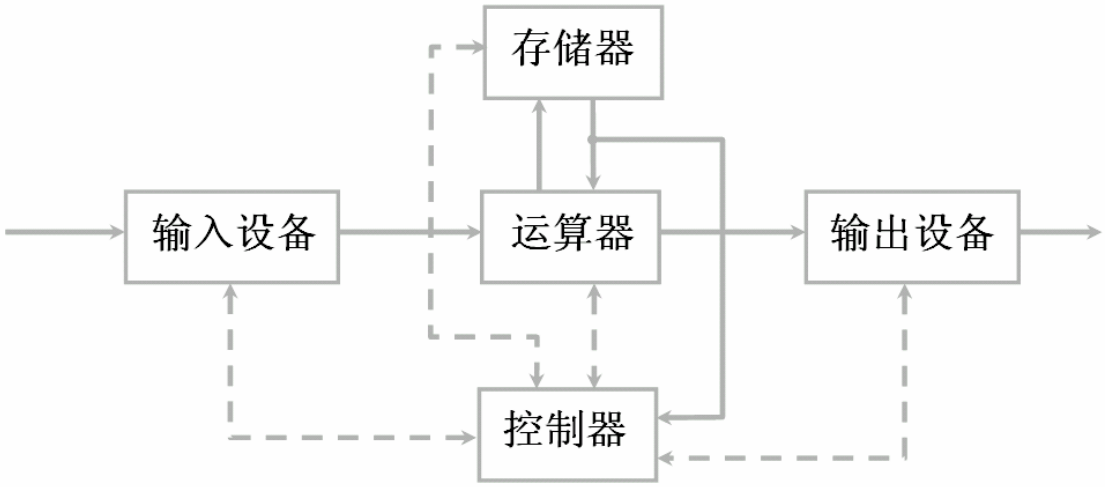
程序设计语言

学习目标：

- 了解程序设计语言对计算的抽象
- 了解程序设计语言的发展

1. 命令式语言（Imperative Languages）

- 冯.诺依曼(Von.Neumann)架构
  - ALU (Arithmetic Logical Unit) + Memory + Input + Output



1. 命令式语言（Imperative Languages）

- 抽象的级别（人机接口）
  - 机器语言（Machine Language）
    - 二进制表示
  - 汇编语言（Assembly Language）
    - 符号（助记符）表示
    - 1950s早期
    - 可重用的宏和子例程

计算7+10并存储的程序

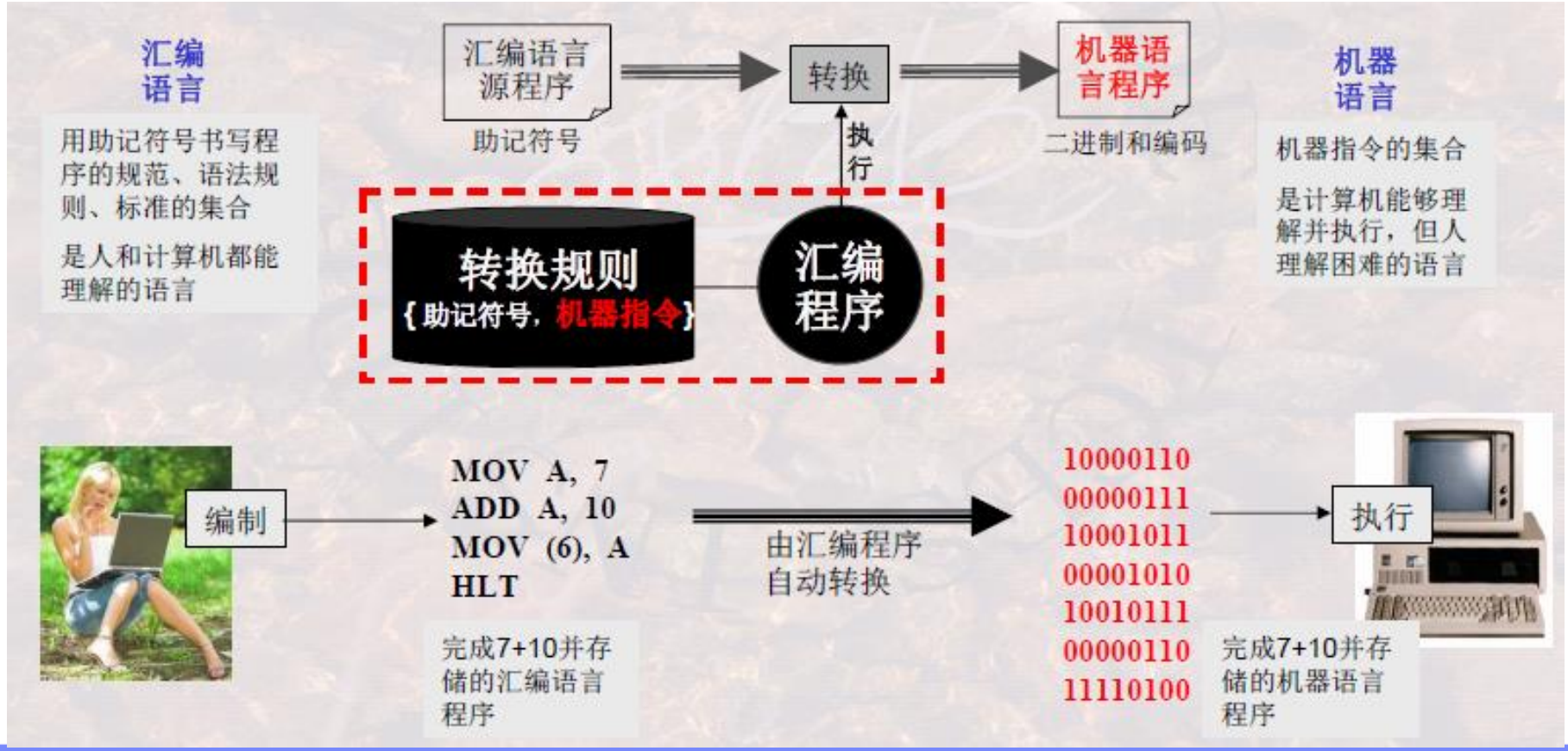
```
10000110 00000111
10001011 00001010
10010111 00000110
11110100
```

计算7+10并存储的程序

```
MOV  A, 7
ADD  A, 10
MOV  (6), A
HLT
```

1. 命令式语言（Imperative Languages）

- 汇编语言程序处理过程



1. 命令式语言（Imperative Languages）

- FORTRAN（FORmula TRANslantion）
  - 由约翰·巴科斯（ John Backus ） 于 1950s 在 IBM 开发
  - 是世界第一个高级语言（High-level Language）
  - 1977获图灵奖，发表了“程序设计能从冯·伊曼形式中解脱出来吗？函数式风格及其程序的代数”（Can Programming be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs）演说
  - 提出了BNF（Backus-Naur Form：用来定义形式语言语法的记号法）
  - 发明了Function-level programming这个概念及实践该概念的FP语言

1. 命令式语言（Imperative Languages）

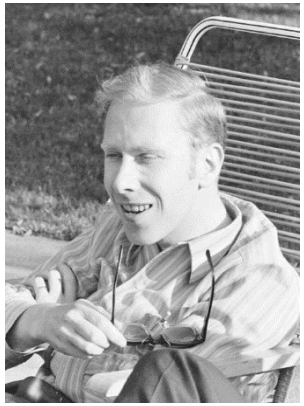
- ALGOL 60（ALGOrithmic Language）
  - 1958-1960设计，Peter Naur获2005年图灵奖
  - BNF(Backus-Naur Form)范式中的N
  - 极大地影响了现代程序设计语言，如块结构：begin … end or {…}、模块化例程、递归例程、变量类型声明、栈存储分配等
  - “Birth of computer science” -- Dijkstra（最短路径算法）
  - “A language so far ahead of its time that it was not only an improvement on its predecessors, but also on nearly all its successors” -- Hoare（快速排序）

1. 命令式语言（Imperative Languages）

- ALGOL 60（ALGOrithmic Language）

```
real procedure average(A,n);
    real array A; integer n;
    begin
        real sum; sum := 0;
        for i = 1 step 1 until n do
            sum := sum + A[i];
        average := sum/n
    end;
```

1. 命令式语言（Imperative Languages）



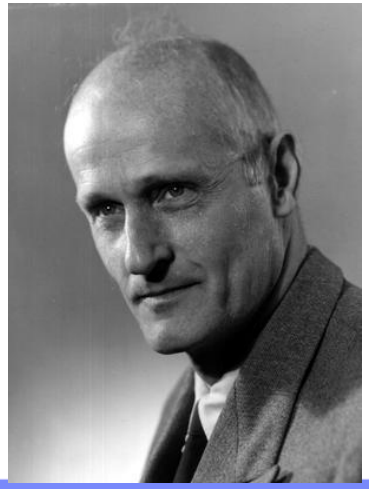
- Pascal
  - 60年代末由Niklaus Wirth设计，1984年图灵奖获得者
  - 修订了Algol的类型系统，比Algol60/68更加严格
  - 流行的教学语言

1. 命令式语言（Imperative Languages）

- BCPL / B / C Family
  - 低级别（low-level）机器访问
    - 内存管理、指针操作
  - 小内存机器的系统编程
    - PDP-7, PDP-11, later VAX, Unix workstations and PCs
    - C has been called a “portable assembly language”
  - 1983年的图灵奖授予Thompson and Ritchie

2. 函数式语言（Functional Languages）

- 命令式程序涉及编写一系列指令，这些指令会改变程序状态以提供解决方案
- 函数式编程将程序视为一系列数学函数
- 基于Turing, Church 和 Kleene 1930's 提出的Lambda Calculus（演算）数学模型



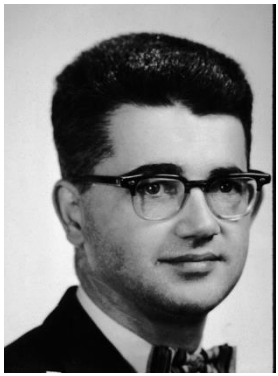
2. 函数式语言（Functional Languages）

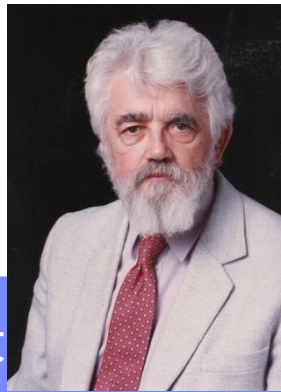
- 命令式编程和函数式编程的区别
  - IP：计算a和b的最大公约数（gcd-Greatest Common Divisor）如果a=b，返回a或b；否则用a和b的差替代其中的最大值并重复进行该操作
  - FP：当a=b时，a和b的gcd定义为a；当a!=b时，a和b的gcd定义为c和d，c为a和b的较小值，d为他们的差

2. 函数式语言（Functional Languages）

- Lisp（List Processing）
  - 由John McCarthy发明（“人工智能”之父，1971年图灵奖获得者）

```
(defun factorial (n)
    (if (<= n 1)
          1
    (* n (factorial (- n 1)))))
```



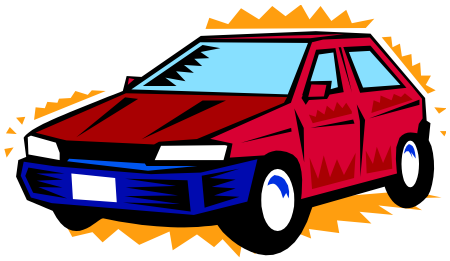


3. 逻辑编程（Logic Programming）

- 把计算看作逻辑推理
  - Gottlob Frege (1848-1925)
  - 基于Horn子句
    - grandfather(x，z)←father(x，y)，father (y，z)
  - Prolog（Programming in logic）1972年由Colmeraner及其研究小组在法国马赛大学提出，适合用来表示人类的思维和推理规则，日本还在其于1979年提出的第五代计算机研究计划中把Prolog列为核心语言

4. 面向对象编程（Object Oriented Programming）

- 数据抽象和隐藏、代码重用
- 第一个面向对象的语言：SIMULA 67
  - Ole-Johan Dahl 和 Kristen Nygaard 一起获得2001年图灵奖
- Simula -> Smalltalk & Ada -> C++ -> Java



车型：法拉利

颜色：红色

年份：1995

活动

发动

停车

加速

5. 其他一些重要的语言

- Algol-like
  - Modula, Oberon, Ada
- 函数式（ Functional ）
  - ISWIM, FP, SASL, Miranda, Haskell, LCF, ML, Caml, Ocaml, Scheme, Common LISP
- 面向对象（ Object-oriented ）
  - Smalltalk, Objective-C, Eiffel, Modula-3, Self, C#, CLOS
- 逻辑编程（ Logic programming ）
  - Prolog, Gödel, LDL, ACL2, Isabelle, HOL

5. 其他一些重要的语言

- 数据处理和数据库（ Data processing and databases ）
  - Cobol, SQL, 4GLs, XQuery
- 系统编程（Systems programming）
  - PL/I, PL/M, BLISS
- 专业应用（Specialized applications）
  - APL, Forth, Icon, Logo, SNOBOL4, GPSS, Visual Basic
- 并发，并行，分布式（Concurrent, parallel, distributed）
  - Concurrent Pascal, Concurrent C, C*, SR, Occam, Erlang, Obliq

5. 其他一些重要的语言

- 编程工具（Programming tool） “mini-languages”
  - awk, make, lex, yacc, autoconf …
- 命令行，脚本，Web（Command shells, scripting and “web” languages）
  - sh, csh, tcsh, ksh, zsh, bash …
  - Perl, JavaScript, PHP, Python, Rexx, Ruby, Tcl, AppleScript, VBScript …
- Web应用程序框架和技术
  - ASP.NET, AJAX, Flash, Silverlight …

思考

- 为什么会有这么多的程序设计语言？
- 哪些因素驱动程序设计语言的发展？
- 程序设计语言有哪些共同点？
- 如何学习一门新的程序设计语言？

计算思维（Computational thinking）

学习目标：

- 了解什么是计算思维
- 了解计算思维的关键技术

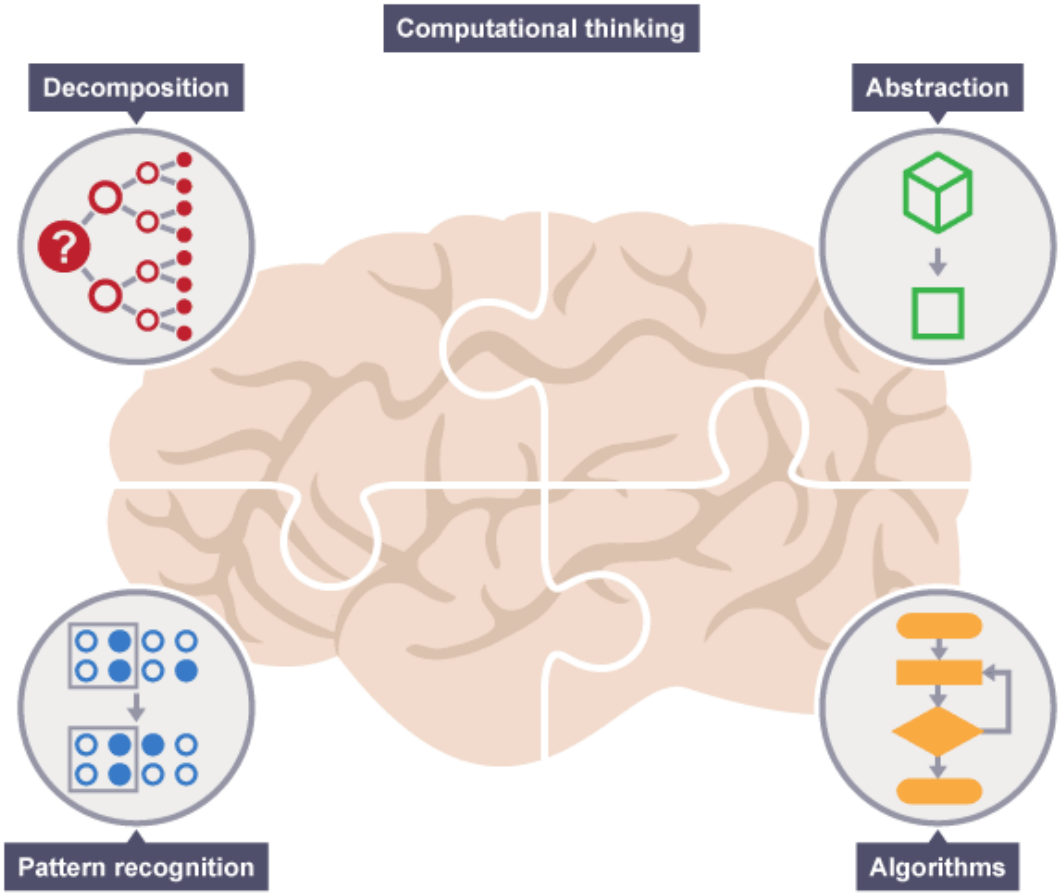
1. 什么是计算思维？

- 计算思维使我们能够解决一个复杂的问题，了解问题所在并开发可能的解决方案。 然后，我们可以以计算机，人类或两者都能理解的方式呈现这些解决方案。

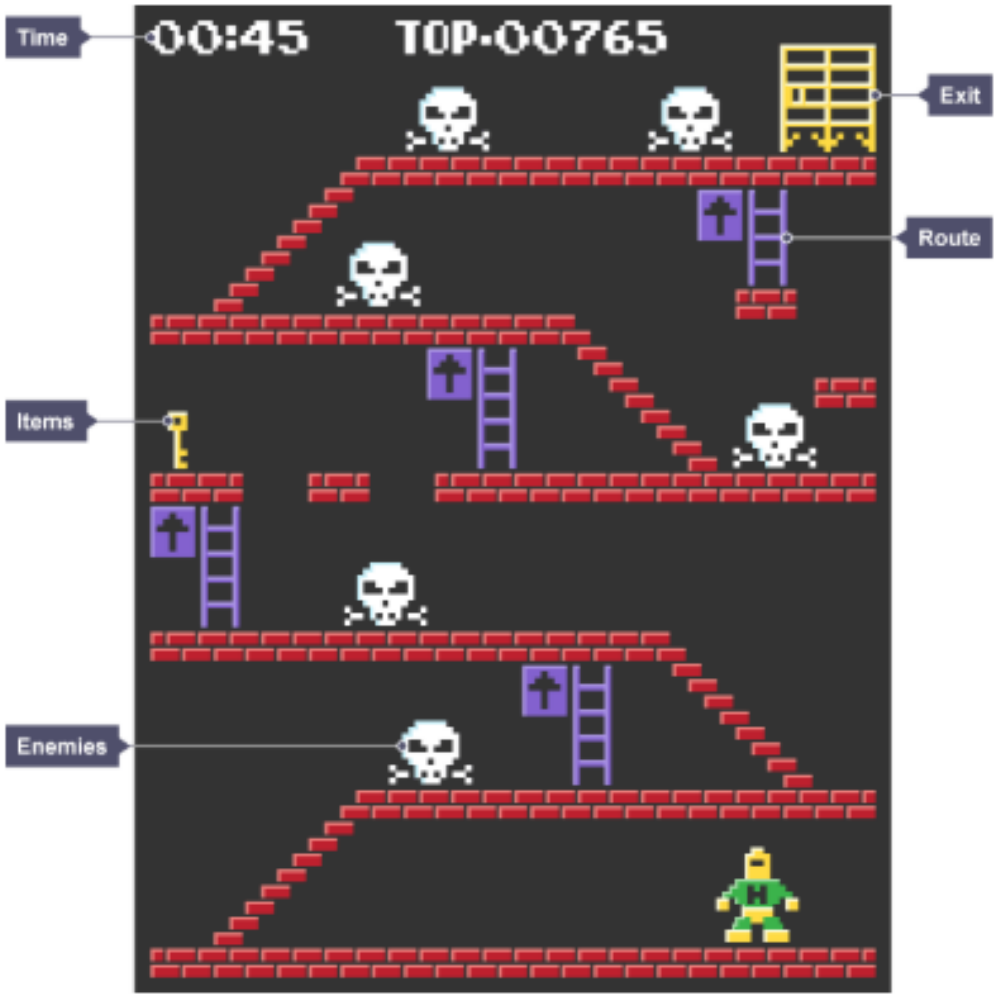
Computational thinking allows us to take a complex problem, **understand what the problem is** and **develop possible solutions**. We can then present these solutions in a way that a computer, a human, or both, can understand.

2. 计算思维的关键技术

- 分解（Decomposition）
- 模式识别（Pattern Recognition）
- 抽象（Abstraction）
- 算法（Algorithms）



2. 计算思维的关键技术

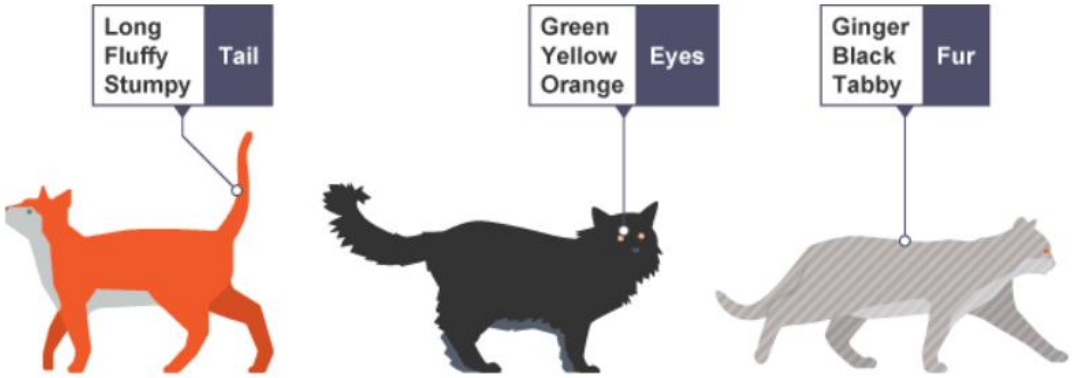


3. 分解（Decomposition）

- 将复杂问题或系统分解为小的可管理且易于理解的部分

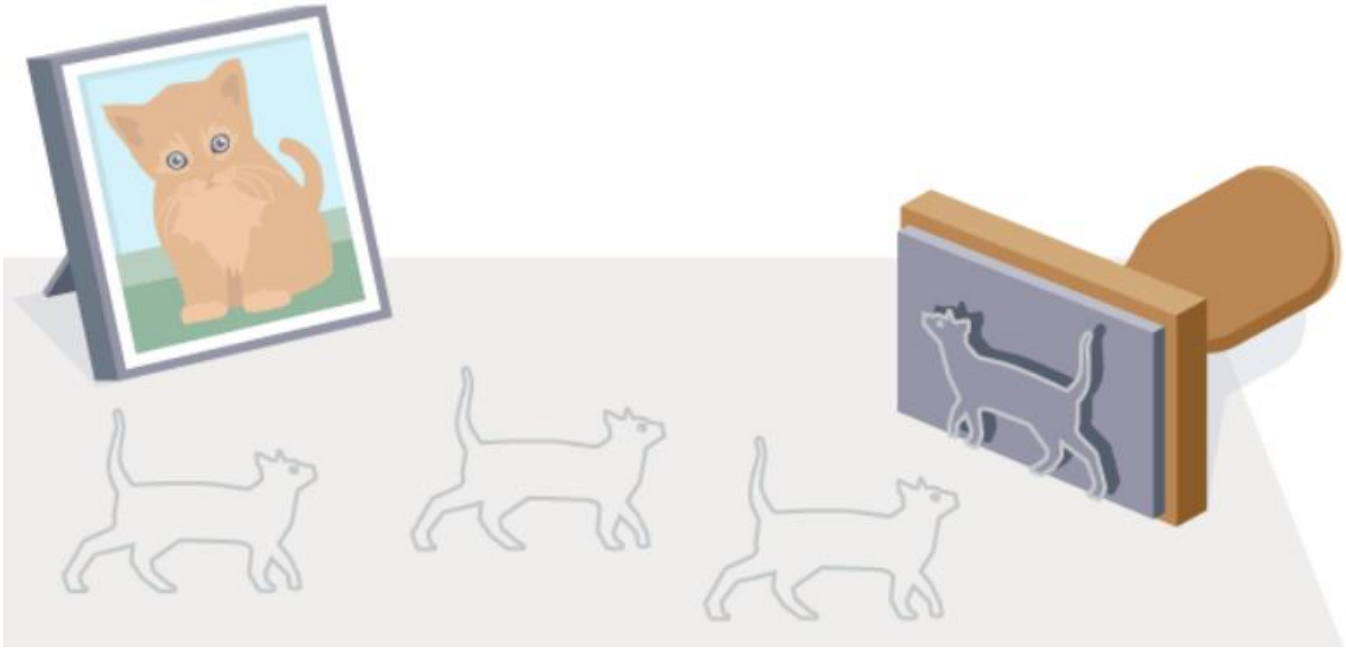
4. 模式识别（Pattern Recognition）

- 找到分解的问题之间的相似性或模式，帮助我们更有效地解决更复杂的问题。



5. 抽象（Abstraction）

- 抽象是一个过滤的过程， 忽略我们不需要的模式特征，以便专注于我们所做的事情。这有助于我们形成对问题的看法。 这个想法被称为“模型”（model）。



6. 算法（Algorithm）

- 算法是一种计划，一组解决问题的指令。算法通常是我们编写程序的起点，它们有时用流程图（flowchart）或伪代码（pseudocode）来表示。

6. 算法（Algorithm）

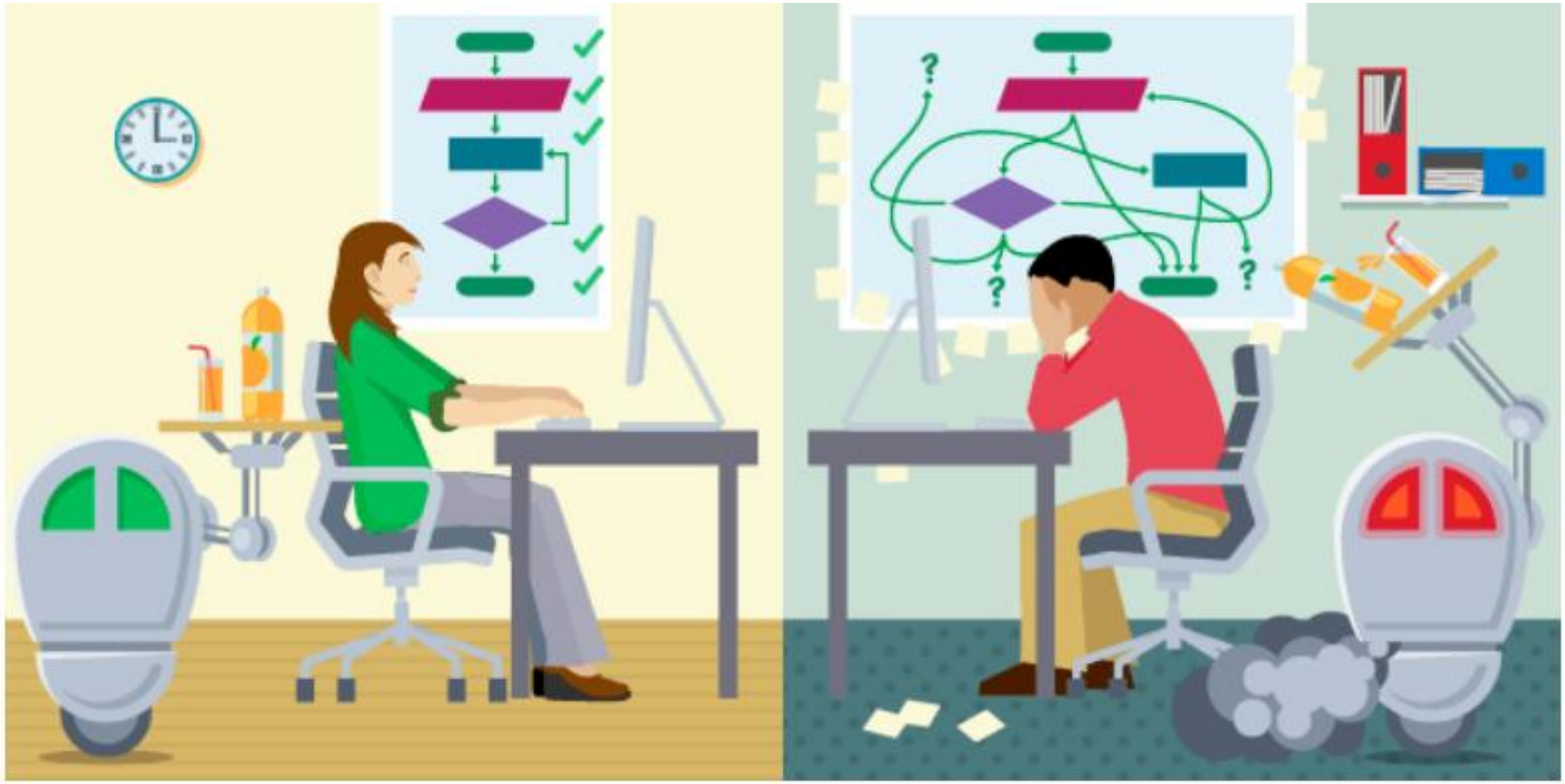
```
OUTPUT 'What is your name?'
INPUT user inputs their name
STORE the user's input in the name variable
OUTPUT 'Hello' + name
OUTPUT 'How old are you?'
INPUT user inputs their age
STORE the user's input in the age variable
IF age >= 70 THEN
        OUTPUT 'You are aged to perfection!'
ELSE
        OUTPUT 'You are a spring chicken!'
```

7. 评估（Evaluation）

- 评估是一个过程，使我们能够确保我们的解决方案能很好的解决所遇到的问题，并思考如何改进它。
  - 是否容易被理解？
  - 是否完整的解决了问题？
  - 是否高效？
  - 是否达到了设计标准？

7. 评估（Evaluation）

- 评估有助于确保在编写解决方案时遇到尽可能少的困难。



小结

- 多种程序设计语言并存，对计算进行不同层次的抽象，解决不同类型的问题
- 计算思维：分解、模式、抽象、算法

作业

- 观看 中国大学MOOC 《软件工程专业导论》1 - 4讲

Crash Course Computer Science

https://www.bilibili.com/video/av21376839/