

第5届全国高等学校计算机程序设计课程研讨会
厦门大学、清华大学出版社联合承办
2015年12月4日-6日 厦门大学



長春工業大學
CHANGCHUN UNIVERSITY OF TECHNOLOGY

CCUT

长春工业大学

王红梅教授

2015年12月

对“程序设计基础”课程的 教学思考

想要探讨的问题

21天学通C语言

C语言三日通

写给大家看的C语言

明解C语言

C语言从入门到精通

C primer plus

C语言入门经典

C语言程序设计：现代方法

啊哈，C!

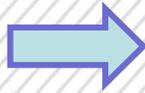
C语言入门很简单

C语言入门1.2.3

1. 作为大学的一门课程，第一门重要的语言基础课程，
目标？教材？过程——教法？学法？考法？

想要探讨的问题

C程序设计语言



C语言程序设计

程序设计基础

C程序设计基础

C程序设计导引

程序设计基础——思想与方法

C程序设计思想与方法

C程序设计教程——设计思想与实现

2. 语言的语法知识与程序设计的思想和方法——
什么是思想？什么是程序设计思想？淡化？强化？

想要探讨的问题

C程序设计新思维

程序员思维修炼

程序设计中常用的计算思维方式

高级语言程序设计——基于计算思维能力培养

程序设计思想与方法——问题求解中的计算思维

C程序设计的抽象思维

C程序设计与计算思维

3. 程序设计语言、程序设计与计算思维——
什么是思维？什么是计算思维？如何理解？如何培养？

想要探讨的问题

C程序设计案例教程

C程序设计实践教程

C程序设计项目教程

C语言程序设计实训教程

C程序设计任务驱动式教程

C程序设计与应用

C语言程序开发范例宝典

C程序设计案例与提高

C程序设计教学做一体化教程

如何编写C程序

4. 学习程序设计语言的目的是求解问题——
初学程序者要解决什么问题？解决问题的目的是什么？

教学对象——一切一切的出发点



从入口到出口，距离有多远！？

以学生为中心，提出实际存在的**教学问题**，发现更加正确的认知规律，规划更加合理的教学内容，采用更加适用的教学方法，进行更加科学的教学设计

问题一——目标？教材？教法？学法？考法？

◆ **教学定位**：从工程应用型人才的培养目标出发，
从大一学生的认知规律和知识基础出发。

◆ **教学效果**：程序设计的渐进层次——

☑ 会写程序（基本的教学效果）

什么叫会写程序？

☑ 会快速地写程序（代码量与编程速度成正比）

☑ 会写高效的程序（高级的教学效果）

☑ 会设计算法（我们一直在努力）

☑ 会改进或发明新算法（发明算法是最高境界）

问题———目标？教材？教法？学法？考法？

◆ 掌握C语言的基本语法和编程环境

语言是（与计算机交流的）工具，工欲善必先利其器

◆ 掌握程序设计的基本思想和一般方法

从计算机的角度设想问题求解的操作步骤

◆ 培养良好的编程风格和一定的程序设计能力

能够应用程序设计语言解决（简单的）实际问题

◆ 培养初步的计算思维能力和算法设计能力

计算思维——模型化、形式化、逻辑思维、抽象思维

算法——程序的灵魂

问题——目标？教材？教法？学法？考法？

第1学期

第5学期

清华大学出版社

8.2 插入排序

直接插入排序过程示例

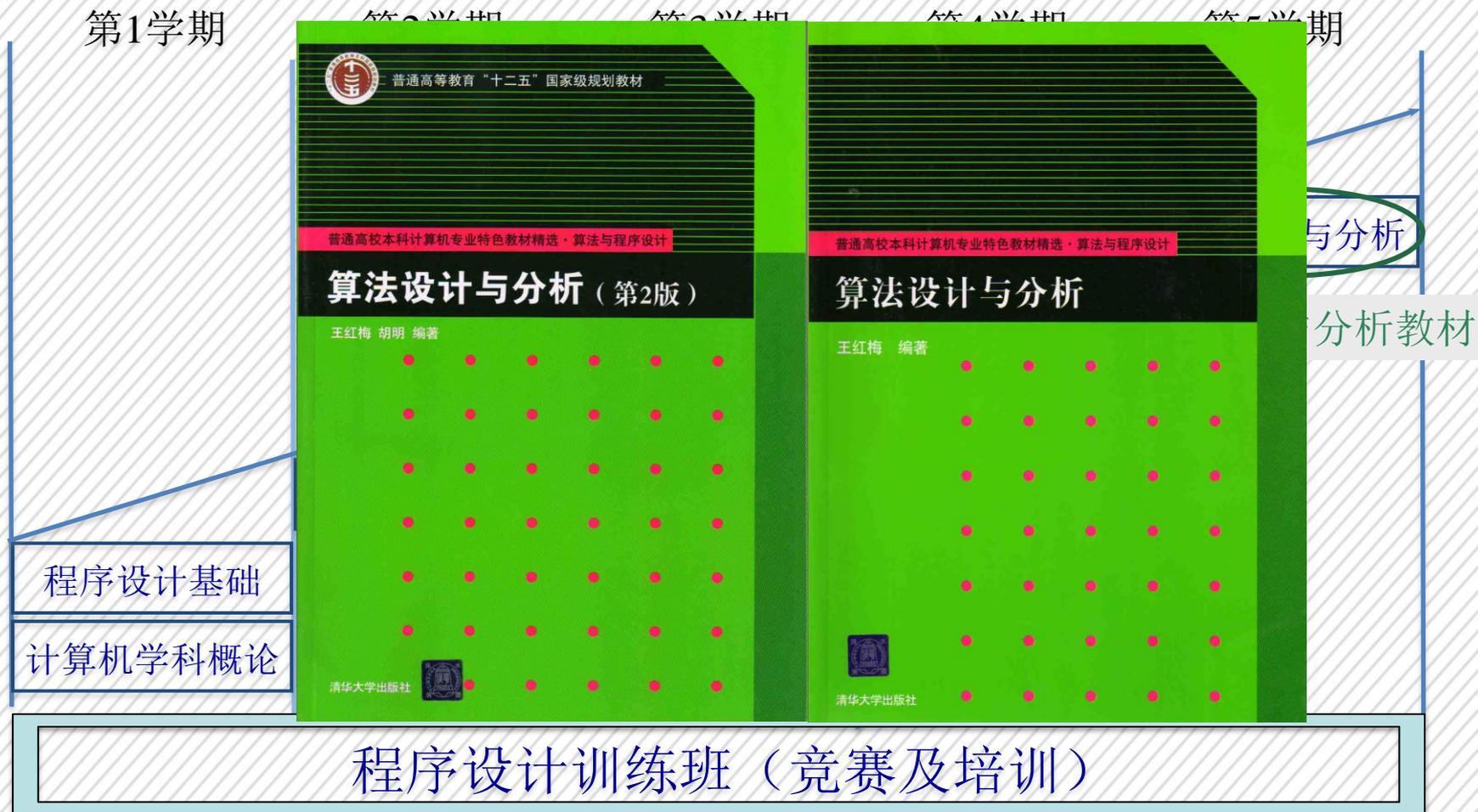
r	0	1	2	3	4	5	6
		21	25	22	10	25*	18
i = 2		21	25	22	10	25*	18
i = 3	22	21	25	22	10	25*	18
i = 4	10	21	22	25	10	25*	18
i = 5	25	10	15	21	25	25*	18
i = 6	18	10	15	21	25	25*	18
		10	15	18	21	25	25*

清华大学出版社

计算机教材

数据结构（C++版）：教育部精品教材，“十一五”、“十二五”国家级规划教材，累计重印50余次，超过100所院校使用

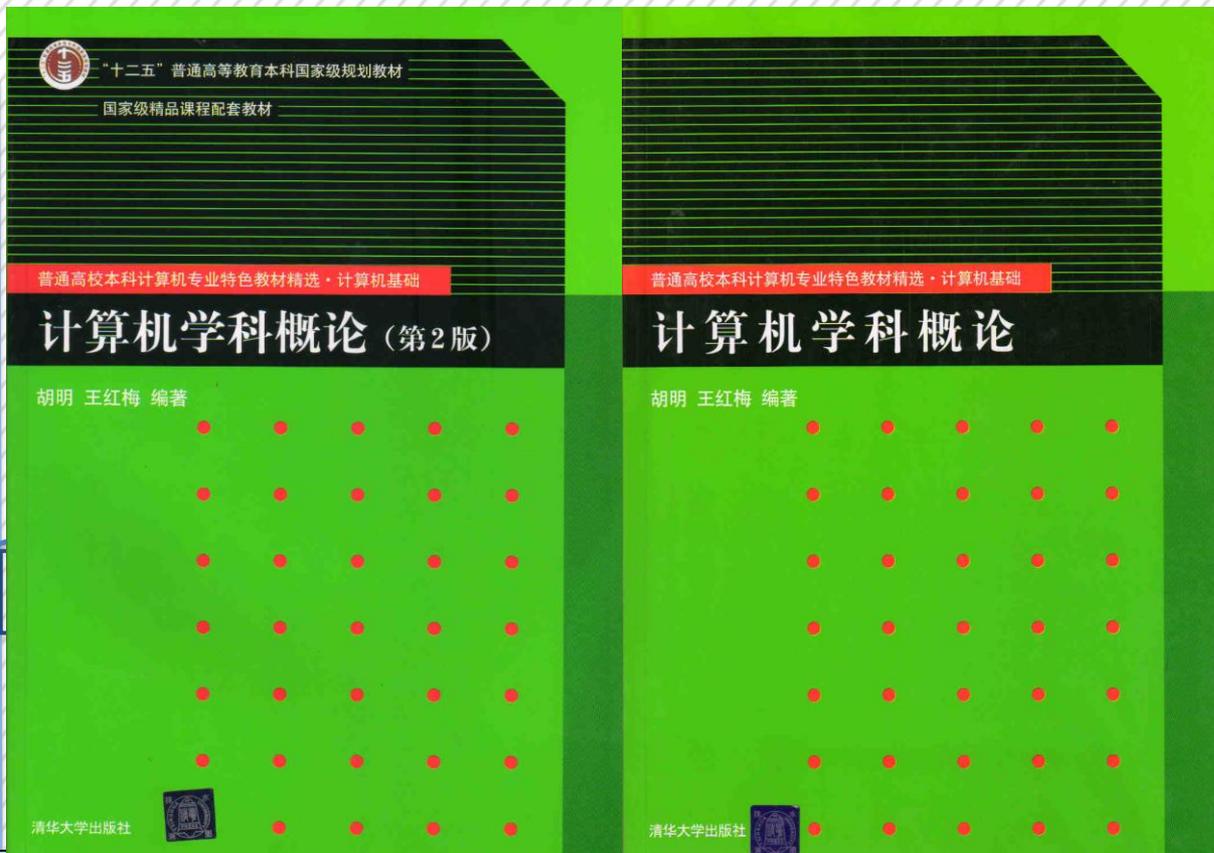
问题——目标？教材？教法？学法？考法？



算法设计与分析（第2版）：“十二五”国家级规划教材，累计重印20余次，超过50所院校使用

问题——目标？教材？教法？学法？考法？

第1学期



分析

程序设计基础

计算机学科概论

程序设计训练班（竞赛及培训）

计算机学科概论（第2版）：国家精品课，“十二五”国家级规划教材，累计重印10余次，超过30所院校使用

问题———目标？教材？教法？学法？考法？

第1学期



程序设计训练班（竞赛及培训）

程序设计基础——从问题到程序（第2版）：“十二五”国家级规划教材，重印5次，超过10所院校使用

问题——目标？教材？教法？学法？考法？

从后续课看，《程序设计基础》教学中存在的问题：

1. 不会写函数，把所有代码都放到main中；
2. 不会写函数头→不会分析问题、不会写递归程序；
3. 不会用指针，基本用法：传参数、申请动态数组；
4. 不会使用数组和结构体，用指针操作数组和结构体；
5. 编码规范，命名、语句框架、程序逻辑。。。

原因？

教材——教学的根本载体！

没有理解基本原理，没有成为基本技能，不用就忘了！

问题——目标？教材？教法？学法？考法？



概念的描述理解

对教学方法的运用

实例的选择运用

对教学意图的融入

知识的前后衔接

对教学目的的实现

内容的组织结构

对教学内容的理解

教学内容 ← 教学对象 + 教学目标

◆ 掌握C语言的基本语法和编程环境

- ☑ **强调基础**：提炼基础性内容，让学生在这门课程涉及的相关领域内，打下终身学习的知识基础——基本概念、基本原理一定要**讲透**，学生才能深刻**理解**。
- ☑ **强调使用**：无须背诵的原则，常用的用着用着就记住了，不常用的，讲了背了考了不用也会忘掉——**讲过的就一定要用**，避免讲是一回事，用是另一回事。
- ☑ **不究细节**：不深究语言细枝末节的语法细节和语言特性，不探究高难的程序设计技巧（有能力的学生除外），但应该给出一些**经典用法**。（类比小孩初学说话）

教学内容——课程的内容设计

一个不容忽视的问题——初学者的知识基础

- ☑ 程序设计需要具备一定的计算机软硬件基本知识，如二进制、内存、整数的补码表示、实数的浮点表示、计算机的工作过程、程序编译的基本过程等，需要融合进来，解决知识的衔接问题——安排在哪儿？怎么讲？
- ☑ 一个原则——**学以致用**，在哪用到就在哪讲，用多少讲多少，讲了就用。（例如补数→补码）

教学内容——课程的内容设计

第1章 绪论

- 1.1 程序、程序设计与程序设计语言
 - 1.1.1 程序与程序设计
 - 1.1.2 程序设计语言
- 1.2 程序的基本构成
 - 1.2.1 程序示例
 - 1.2.2 基本字符集
 - 1.2.3 词法单位
 - 1.2.4 语法单位
 - 1.2.5 程序
- 1.3 程序的上机过程
 - 1.3.1 编程环境
 - 1.3.2 程序编辑
 - 1.3.3 程序编译
 - 1.3.4 程序连接
 - 1.3.5 运行调试
- 1.4 程序风格
 - 1.4.1 标识符的命名规则
 - 1.4.2 注释
 - 1.4.3 缩进
 - 1.4.4 行文格式
- 1.5 问题求解与程序设计
 - 1.5.1 程序设计的一般过程
 - 1.5.2 程序设计的核心——算法
 - 1.5.3 程序设计实例——鸡兔同笼问题

第2章 数据的存储表示

【引例2.1】计算圆的面积

- 2.1 数据的存储
 - 2.1.1 二进制
 - 2.1.2 内存（地址、存储单元、存储内容）
 - 2.1.3 冯诺伊曼体系结构的核心——存储程序
 - 2.2 基本数据类型（存储格式、取值、运算）
 - 2.2.1 整型（补码）
 - 2.2.2 实型（浮点表示）
 - 2.2.3 字符型（ASCII）
 - 2.2.4 逻辑型
 - 2.3 常量
 - 2.3.1 字面常量
 - 2.3.2 符号常量
 - 2.4 变量
 - 2.4.1 变量的概念（从内存的角度讲透）
 - 2.4.2 变量的定义和初始化
 - 2.4.3 变量的赋值
- ## 第7章 变量的间接访问——指针
- 7.1 指针
 - 【引例7.1】获取密电码
 - 7.1.1 指针的概念（从内存的角度讲透）

教学内容——课程的内容设计

一个客观存在的问题——语言的结构不良性

- ☑ 语言的结构不良性导致**不能完全按照知识的结构顺序推进**，科学安排知识单元之间的拓扑关系，对于有一定难度的主题采用**增量**的方式。
- ☑ 第一节课给出两个程序的例子（用自然语言和文章做类比），带领学生读程序，找规律，引出并**标识**基本概念。
- ☑ 对于printf函数和scanf函数，只要求掌握%d、%f（%lf）、%c，**讲的越多越糊涂**，讲过不用很快就忘掉了，在学生熟练使用常用格式后，讲到库函数时再给出全部说明。

教学内容——课程的内容设计

一个客观存在的问题——语言的结构不良性

函数

- 自定义函数
- 库函数
- 变量的作用域
- 变量的生存期



再谈函数

- 函数的嵌套调用
- 函数的递归调用

指针

- 指针的概念及操作
- 指针作为函数的参数
- 数组作为函数的参数



再谈指针

- 指针与数组
- 指针与结构体
- 动态存储分配

输入/输出

- 输入/输出的概念
- 输入/输出函数



再谈输入/输出

- 文件的概念
- 文件的读写操作

程序的基本结构

- 单文件程序
- 自定义函数
- 文件包含



再谈程序的基本结构

- 多文件程序
- 外部变量和外部函数
- 嵌套包含

教学内容——课程的结构设计

基本篇

提炼基础性知识

构造篇：数据组织

数据的存储表示

基本数据类型

- 整型
- 实型
- 字符型
- 逻辑型

数据的基本表现形式

- 常量
- 变量

数据的运算处理

数据的基本运算

- 算术运算
- 逻辑运算
- 赋值运算
- 其他运算
- 运算对象的类型转换

变量的访问方式

变量的直接访问

- 存操作
- 取操作

变量的间接访问

- 指针的概念
- 指针变量的操作
- 指针所指变量的操作

程序的流程控制

程序的基本控制结构

- 顺序结构
- 分支结构
- 循环结构

程序的组装单元——函数

- 自定义函数
- 系统库函数
- 变量的作用域
- 变量的生存期

数组

- 一维数组
- 二维数组

字符串

- 字符数组
- 字符串指针

自定义数据类型

- 枚举类型
- 结构体类型

结构体数组

再谈篇

再谈函数

- 函数的嵌套调用
- 函数的递归调用

再谈指针

- 指针与数组
- 指针与结构体
- 动态存储分配

再谈输入/输出——文件

- 文件与文件指针
- 文件的位置指针
- 文件的打开关闭
- 文件的读写操作

再谈程序的基本结构

- 多文件程序
- 外部变量
- 外部函数
- 嵌套包含

低级篇

低级程序设计

- 数据表示
- 位运算

教学内容——课程的结构设计

基本篇

数据的存储表示

基本数据类型

- 整型
- 实型
- 字符型
- 逻辑型

数据的基本表现形式

- 常量
- 变量

函数放到哪儿？指针讲什么？
避免讲了不用 → 不会用、后续课

数据的基本运算

- 算术运算
- 逻辑运算
- 赋值运算
- 其他运算
- 运算对象的类型转换

- 存操作
- 取操作

变量的间接访问

- 指针的概念
- 指针变量的操作
- 指针所指变量的操作

- 指针作为函数的参数
- 数组作为函数的参数

程序的流程控制

程序的基本控制结构

- 顺序结构
- 分支结构
- 循环结构

程序的组装单元——函数

- 自定义函数
- 系统库函数
- 变量的作用域
- 变量的生存期

构造篇：数据组织

数组

- 一维数组
- 二维数组

字符串

- 字符数组
- 字符串指针

自定义数据类型

- 枚举类型
- 结构体类型

结构体数组

再谈篇

再谈函数

- 函数的嵌套调用
- 函数的递归调用

再谈指针

- 指针与数组
- 指针与结构体
- 动态存储分配

再谈输入/输出——文件

- 文件与文件指针
- 文件的位置指针
- 文件的打开关闭
- 文件的读写操作

再谈程序的基本结构

- 多文件程序
- 外部变量
- 外部函数
- 嵌套包含

低级篇

低级程序设计

- 数据表示
- 位运算



教学设计——原理为纲

一个看似简单的问题——语法怎么讲？

问题1：割裂了语法和语义。

【语法】变量定义的一般形式如下：

类型说明符 变量名列表；←—— 以分号结尾

其中，类型说明符是任意合法的数据类型，包括基本数据类型和自定义数据类型；变量名列表是一个变量名或由逗号分隔的多个变量名；最后用分号表示结束变量定义。

【语义】将变量名列表的各个变量定义为类型说明符的类型，编译器为各变量分配相应的存储单元，并将变量名与这个存储单元绑定在一起。

教学设计——原理为纲

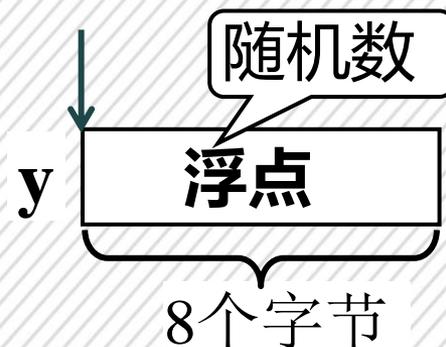
一个看似简单的问题——语法怎么讲？

问题2：语义一定要站在内存的角度讲。

```
int x;          /*定义x为整型变量*/  
double y;      /*定义y为双精度实型变量*/
```



```
int x = 5;  
printf(“%3.1f”, x);  
double y = 2.5;  
printf(“%d”, y);
```



数据类型对初学者很重要！是跨越抽象的关键！

变量的属性：**数据类型**（X个字节、存储格式、取值范围，运算集合）、变量名、变量地址（存储单元）、变量值。

教学设计——原理为纲

一个看似简单的问题——语法怎么讲？

问题3：造句——文章，语句——程序。

例 4.3 求两个整数中的较大值。

解：设整数 x 和 y 的较大值为 \max ，可以先假定 x 较大，再判断 \max 是否小于 y ，如果 \max 小于 y ，则较大值为 y 。语句如下：

```
max = x ;
```

```
if (max < y)
```

```
    max = y ;
```

```
/*假设 x 的值较大*/
```

```
/*如果 max 小于 y，则较大值是 y*/
```

教学设计——原理为纲

用简单原理解释看似复杂

2.4 变量

2.4.1 变量的概念（从内存的角度讲透）

2.4.2 变量的定义和初始化

2.4.3 变量的赋值

第5章 批量同类型数据的组织——数组

5.1 一维数组

【引例5.1】舞林大会

5.1.1 一维数组的定义和初始化

5.1.2 一维数组的操作（输入/输出、赋值、运算）

5.2 二维数组

【引例5.2】哥尼斯堡七桥问题

5.2.1 二维数组的定义和初始化

5.2.2 二维数组的操作（输入/输出、赋值、运算）

第6章 程序的组装单元——函数

6.1 用户定义的函数——自定义函数

【引例6.1】欧几里德算法（函数版）

6.1.1 函数定义

6.1.2 函数调用（形参变量的初始化、赋值）

【引例9.1】整数中的重复数字

9.1.1 枚举类型的定义

9.1.2 枚举变量的定义与初始化

9.1.3 枚举变量的操作

9.2 不同类型数据的组织——结构体类型

【引例9.2】统计入学成绩

9.2.1 结构体类型的定义

9.2.2 结构体变量的定义和初始化

9.2.3 结构体变量的操作

9.3 批量不同类型数据的组织——结构体数组

【引例9.3】统计入学成绩（改进版）

9.3.1 结构体数组的定义和初始化

9.3.2 结构体数组的操作

第7章 变量的间接访问——指针

7.1 指针

【引例7.1】获取密电码

7.1.1 指针的概念（从内存的角度讲透）

7.1.2 指针变量的定义和初始化

7.1.3 指针变量的赋值

7.1.4 指针所指变量的间接访问（赋值、运算）

第8章 字符数据的组织——字符串

【引例8.1】恺撒加密

8.1 字符串变量的定义和初始化

8.1.1 字符数组 8.1.2 字符串指针

8.2 字符串的操作

8.2.1 输入/输出操作

8.2.2 赋值操作

教学设计——方法为上

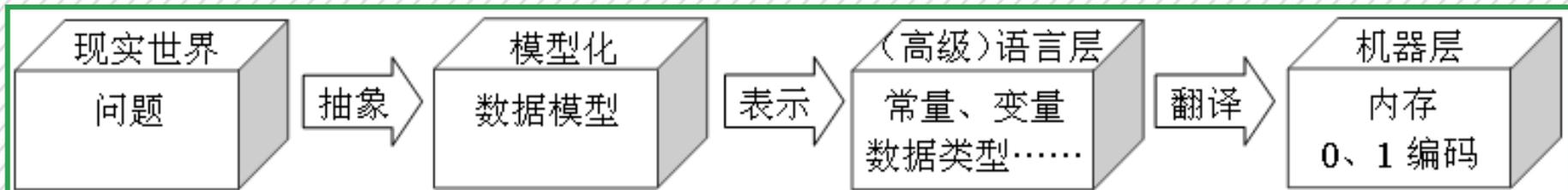
一个不能丢弃的教学方法——手工执行程序

- ◆ 学习程序设计的方法：读程序、写程序。
- ◆ 设断点，**单步执行程序**，观察变量的变化过程。
- ◆ **手工执行程序**：站在计算机内存的角度，通过图示阐述内存（变量）在程序执行过程中的动态变化过程，跨越程序设计的抽象性。
- ◆ 不会手工执行程序（没有手工执行程序的习惯）→不会手工执行算法→**离开计算机不会干活**。
- ◆ **知其然→知其所以然→使其然**，才能很好地驾驭程序设计语言解决实际问题。

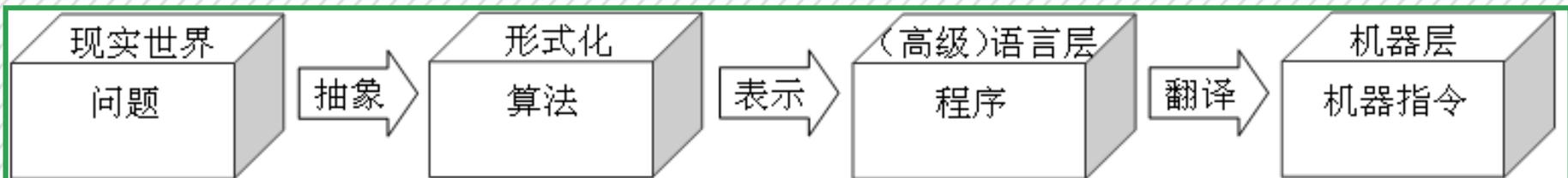
问题二——什么是思想？什么是程序设计思想？

◇ 数据结构 + 算法 = 程序。

◇ 程序设计的关键：**数据表示**和**数据处理**。



数据表示：**抽象**出数据模型，将其从机外表示转换为机内**表示**



数据处理：**抽象**出问题的求解方法，将其转换为程序

问题二——如何体现程序设计思想

- ◇ 作为大学的第一门程序设计课程，无疑将影响思维方式——**正确的程序设计方法**。
- ◇ 大学课程本来是为了探讨问题而开设的，问题的核心内容——**思考问题的习惯**。

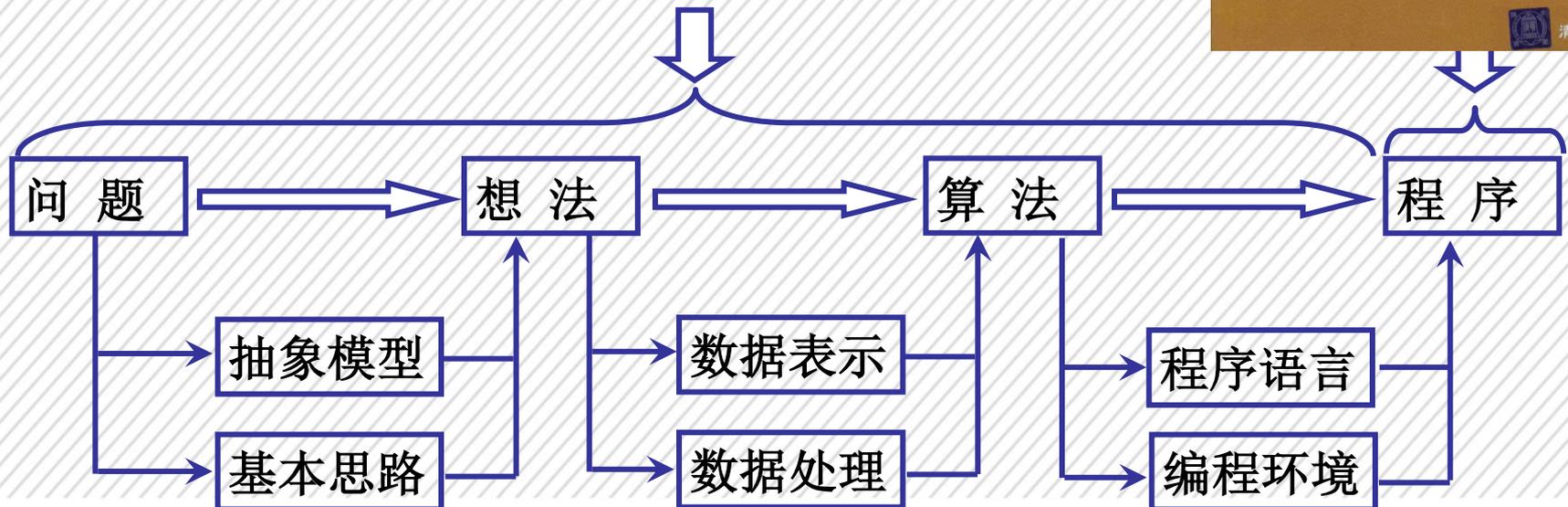
高等学校计算机基础教育规划教材

程序设计基础 ——从问题到程序

胡明 王红梅 编著

清华大学出版社

人（设计方案）



问题二——如何体现程序设计思想？

语言是交流的工具，学会用程序设计语言**说话**

◆ 语法**知识** \leftrightarrow 程序设计**能力**

◆ 学会了语言的语法知识，知识会忘掉；掌握了程序设计的能力，能力会形成一种基本技能、一种本能。

◆ **语法**：写语句，强调基本语法的理解和使用。

◆ **问题**：写程序，按照“问题 \rightarrow 想法 \rightarrow 算法 \rightarrow 程序”的一般过程，强调程序设计的思想和方法。

在问题求解的过程中，体会**数据表示**的作用，学会将数据从机外表示转化为机内表示——**常量、变量、数据类型**；体会**数据处理**的作用，学会用**伪代码描述想法**——**会写算法**，学会用语句描述求解问题的操作步骤——**会写程序**。

问题二——如何体现程序设计思想？

一个有争议的话题：简单程序要不要写算法？

鸡兔同笼问题

$$\begin{cases} x + y = M \\ 2x + 4y = N \end{cases} \quad 0 \leq x, y \leq M$$

伪代码

step1: chicken从0 ~ M重复执行下述操作：

step1.1: rabbit = M - chicken;

**step1.2: 如果(2 * chicken + 4 * rabbit等于N) ,
则跳出循环;**

step1.3: chicken++;

**step2: 如果是提前跳出循环，则输出chicken和rabbit的值;
否则输出“无解”;**

问题二——如何体现程序设计思想？

一个有争议的话题：简单程序要不要写算法？

◆ 写程序之前一定要写算法，至少在头脑中过一遍，培养的是一种**思维习惯**。

◆ **写算法的好处**：算法训练就像一种**思维体操**，能够锻炼思维，引导思维过程，培养抽象思维和逻辑思维能力，使思维变得更清晰、更有逻辑。

◆ “问题→想法→算法”是与程序设计语言无关的；

◆ 程序设计的基本思想和方法——从**计算机的角度**（**构造性的思维**）设想问题求解的操作步骤。

◆ 理解**形式化**，培养**逻辑思维**和**抽象思维**。

问题二——如何体现程序设计思想？

一个有争议的话题：简单程序要不要写算法？

- ◆ 算法训练的**副作用**：有些学生自认智商不够，对算法产生畏难心理，对程序设计失去兴趣。
- ◆ 说明算法的层次：
 - ☑ 发明算法是计算机学者的**最高境界**。
 - ☑ 写算法可以培养人的思维，思维的重要性不言而喻。
 - ☑ 凡事都是从简单开始，区别就在于是否用心、能否坚持。
 - ☑ 以后可以不搞算法，但现在不能不练算法。

问题二——如何体现程序设计思想？

一个不用争议的话题：用什么来描述算法？

- ◆ 流程图是**out**的，伪代码是**in**的。伪代码的**抽象分级**。
- ◆ 《程序设计基础》要教会学生把想法用**伪代码**描述出来。
- ◆ 循环（构造性思维）的几种写法：

伪代码

step2: 当x不等于0时，重复执行下述操作：

。 。 。

step2: 重复执行下述操作，直到x等于0：

。 。 。

step2: 循环变量 i 从 1 ~ x，重复执行下述操作：

。 。 。

问题三——什么是计算思维？如何理解？

一个问题：初学程序者要不要掌握一种思维模式？

◆ 任何一种思维的形成，都需要有**一定的模式**，都需要**大量的练习**，在潜移默化中形成。

◆ 思维培养是一个综合性的系统工程，把计算思维的培养构建一个教学体系，需要思考如下问题：

计算思维的特征和表现是什么？

计算思维有哪些组成？

各组成部分在哪些课程中讲授？

如何将其贯彻到具体课程的知识单元中？

???

问题三——什么是计算思维？如何理解？

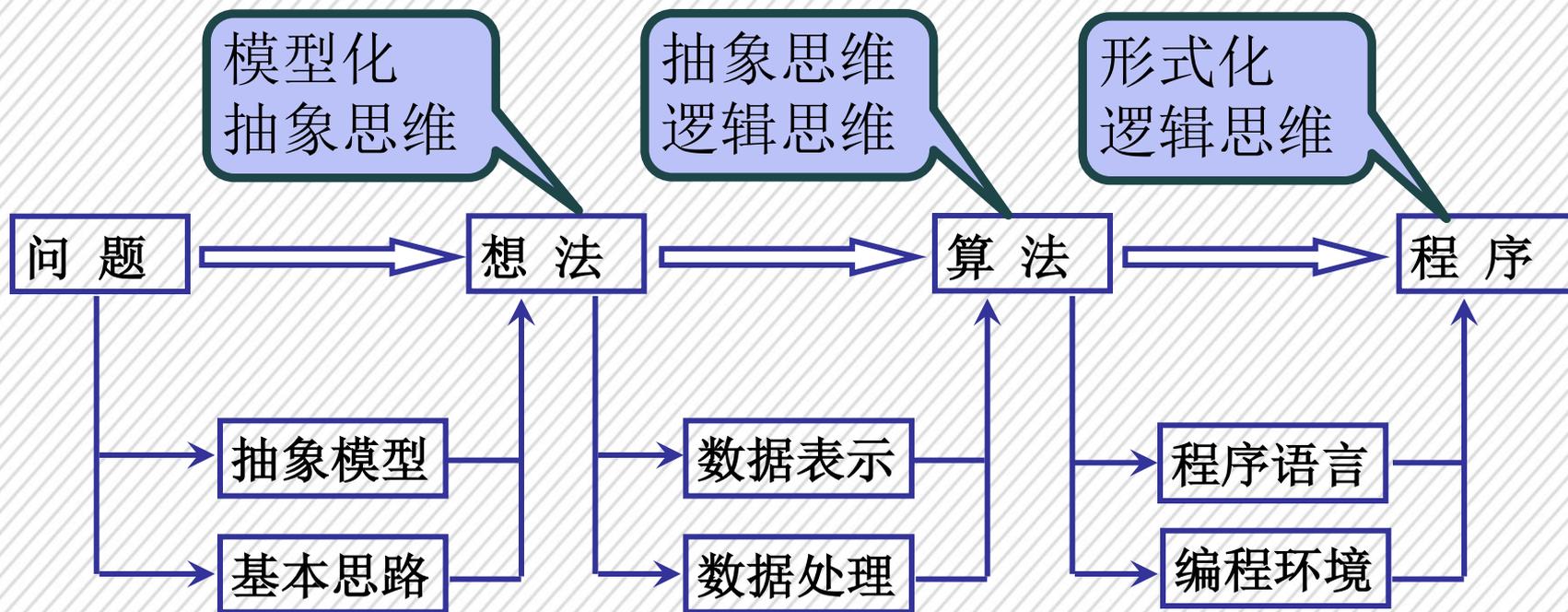
- ◆ 程序设计关注的是从提出问题（面对问题）到抽象出解决问题的算法，直至最终编写程序解决问题的整个思维过程，这个过程正是**计算思维的运用过程**。
- ◆ 从算法的角度，在没有提出计算思维的时候，人们就已经**计算思维般**地进行计算思维了（一种数学思维？）。
- ◆ 将计算思维的基本特征和方法进行梳理和组织，在程序设计过程中培养和运用。

今有田广七分步之四，从五分步之三。问为田几何？答曰：三十五分步之十二。又有田广九分步之七，从十一分步之九。问为田几何？答曰：十一分步之七。又有田广五分步之四，从九分步之五。问为田几何？答曰：九分步之四。

乘分**术**曰：母相乘为法，子相乘为实，实如法而一。

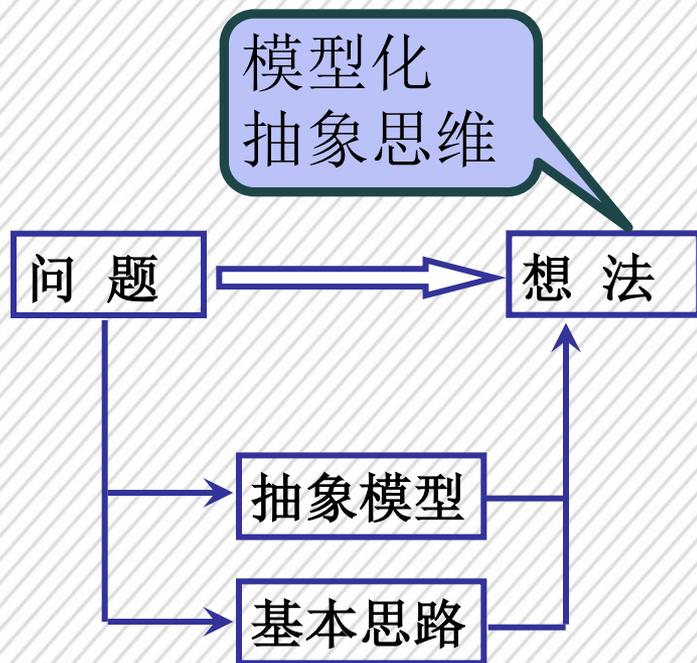
问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维



问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维

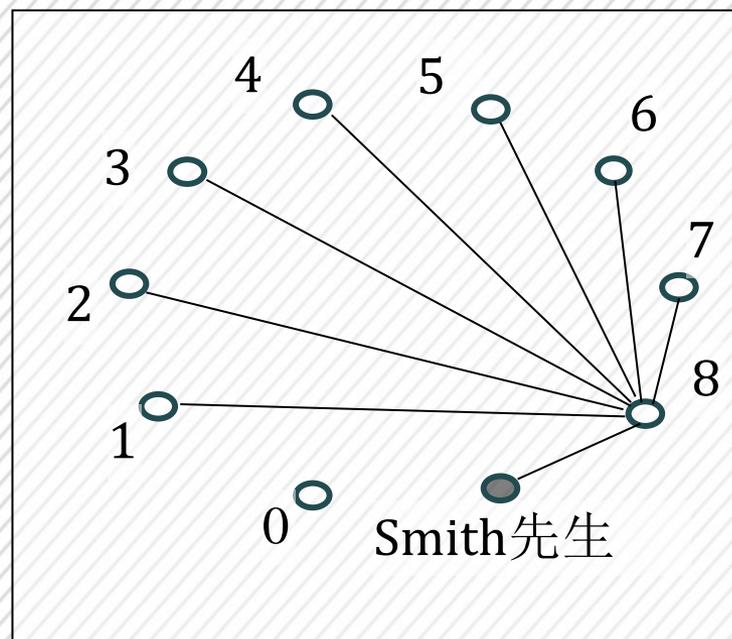
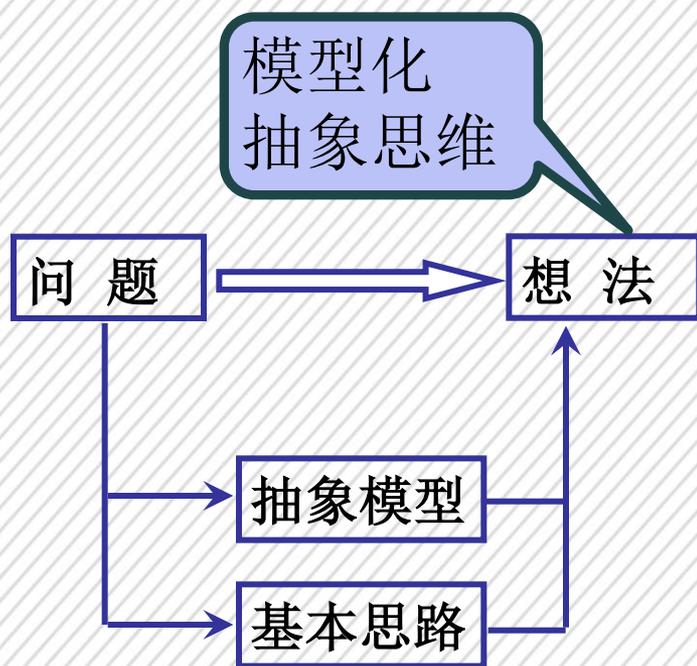


◇ 有些问题难以求解的原因是无从下手，如果能对问题建立一个合适的模型，则问题可能会变得豁然开朗。

◇ **握手问题**。**Smith**先生和太太邀请四对夫妻来参加晚宴。每个人来的时候，房间里的一些人都要和其他人握手。当然，每个人都不会和自己的配偶握手，也不会跟同一个人握手两次。之后，**Smith**先生问每个人和别人握了几次手，他们的答案都不一样。问题是，**Smith**太太和别人握了几次手？

问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维



七桥问题、选课问题。。。

问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维

百元买百鸡问题

$$\begin{cases} x + y + z = 100 \\ 5 \times x + 3 \times y + z / 3 = 100 \end{cases}$$

Fibonacci数列问题

$$f(n) = \begin{cases} 1 & n = 1 \\ 1 & n = 2 \\ f(n - 1) + f(n - 2) & n > 2 \end{cases}$$

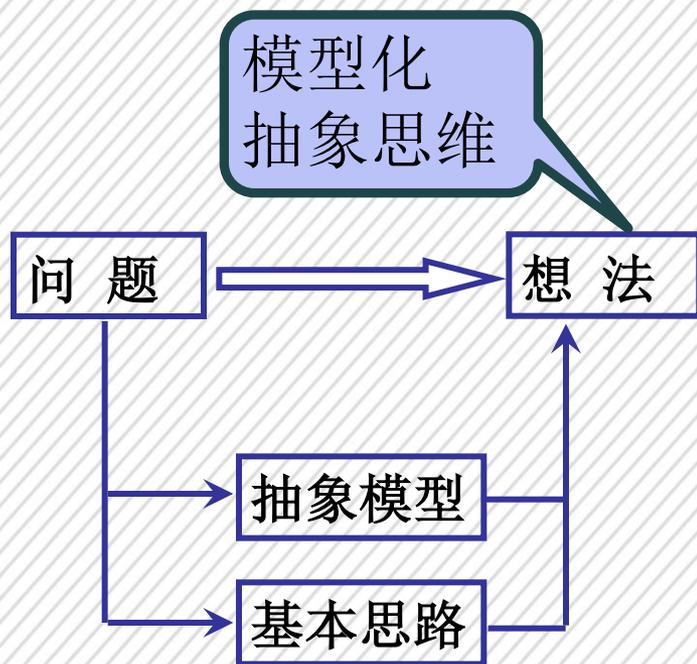
更复杂的问题：**KMP**。。。。

$$\text{next}[j] = \begin{cases} -1 \\ \max \{k \mid 1 \leq k < j \text{ 且 } T[0] \dots T[k-1] = T[j-k] \dots T[j-1]\} \\ 0 \end{cases}$$

$j = 0$
集合非空
其它情况

问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维



◇ **模型**：对一个问题或想法进行形式化、特征化、可视化思维的方法。

◇ **方法**：运行实例，寻找解决问题的基本思路。

◇ **抽象**：问题→模型；实例→想法。

◇ **存在的问题**：学生没有掌握可视化思维的方法（不会动笔）→计算思维应该是一种**普适的思维模式**。

问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维

问题：将三个数由小到大排序。

想法：设三个数分别是 x 、 y 和 z ，先将 x 和 y 进行比较，如果 $x > y$ ，则将 x 和 y 交换；再将 z 和 y 、 x 进行比较，有三种情况：

- (1) 如果 $z \geq y$ ，此时变量 x 、 y 和 z 即为从小到大排列；
- (2) 如果 $x \leq z < y$ ，则将 y 和 z 交换；
- (3) 如果 $z < x$ ，则从小到大依次为 z 、 x 、 y ，4条赋值语句实现。

抽象思维
逻辑思维

想法

算法

数据表示

数据处理

问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维

问题：将三个数由小到大排序。

step1: 如果 $x > y$ ，则将 x 和 y 交换；
step2: 如果 $z < x$ ，则 $temp = z$ ； $z = y$ ；
 $y = x$ ； $x = temp$ ；
否则，如果 $z < y$ ，则将 y 和 z 交换；
step3: 依次输出 x, y, z ；

抽象思维
逻辑思维

想法

算法

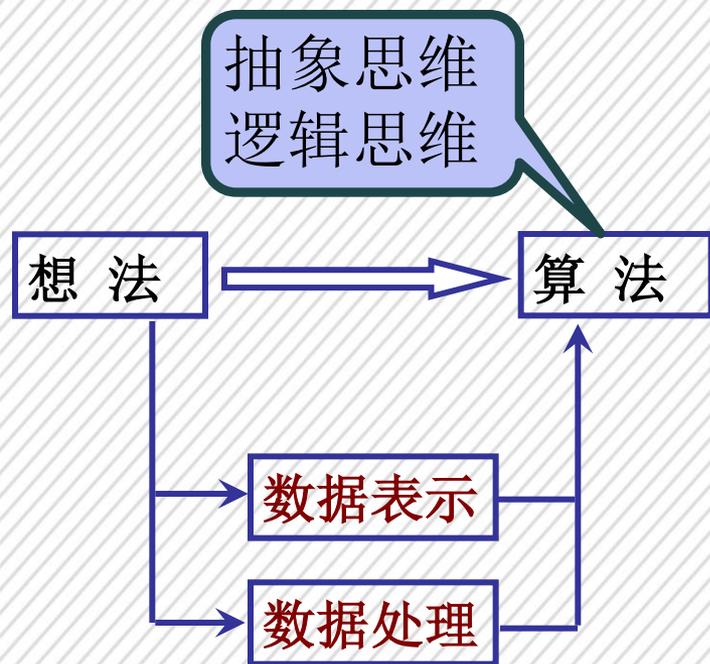
数据表示

数据处理

算法的抽象分级使得思维可以在多个抽象层次上思考问题，显然能提高**思维伸缩**的能力。

问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维



◇ 存在的问题：

(1) **抽象**：通常只有一个混沌的想法就着手编写程序，没有从实例的求解过程把算法抽象出来。

(2) **逻辑**：没有形成良好的逻辑思维能力，稍复杂的逻辑就混乱了。

◇ 解决办法：

(1) 多看多练。能力是练出来的。

(2) 把思维**放慢**。其根本是从计算机的角度思考求解问题的操作步骤。

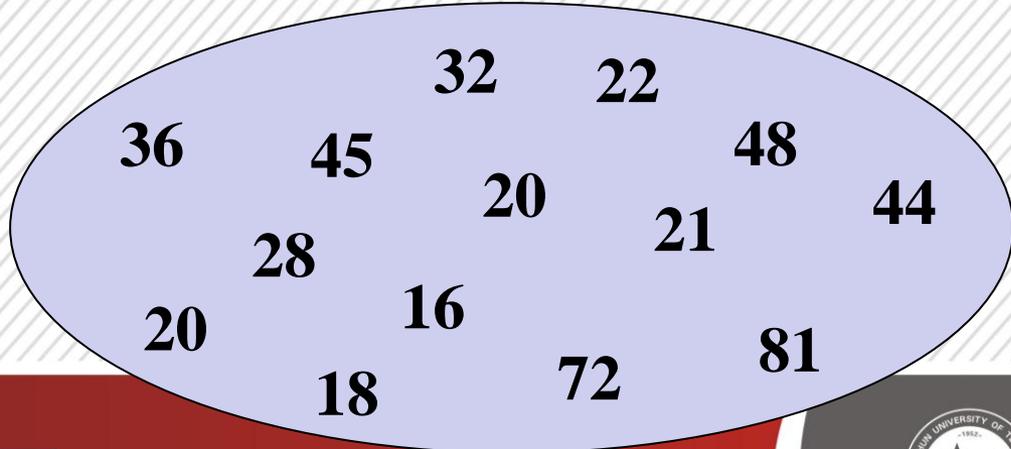
问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维

什么叫把思维**放慢**？如何从计算机的角度思考求解问题的操作步骤？

思考1：如何计算 $3 + 5 \times 2 - 8$ ？

思考2：如何找最大值？



抽象思维
逻辑思维

想法

算法

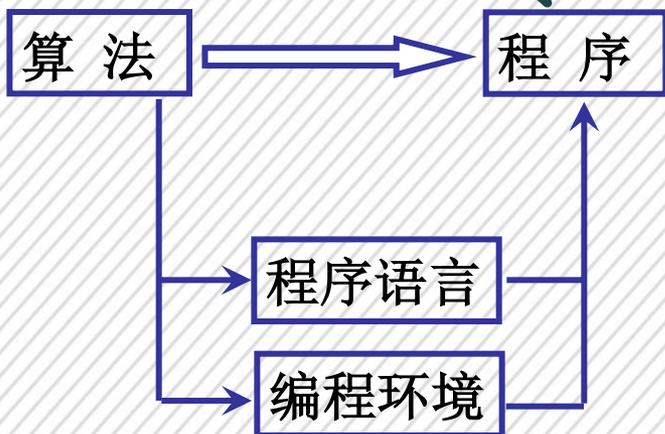
数据表示

数据处理

问题三——如何理解和培养计算思维？

计算思维：模型化、形式化、抽象思维、逻辑思维

形式化
逻辑思维



问题：如下语句中“5”表示什么？

```
int x = 5;
```

体会**数据表示**的含义是什么，符号化的含义是什么，体会**计算的本质**就是符号变换，体会形式化的计算机仅有语法，没有语义。

问题三——如何理解和培养计算思维？

计算思维： 模型化、形式化、抽象思维、逻辑思维

形式化
逻辑思维

算法

程序

程序语言

编程环境

问题1：温度转换。

$$\text{temC} = 5 * (\text{temF} - 32) / 9$$



$$\text{temC} = 5 / 9 * (\text{temF} - 32)$$



问题2：字符串匹配。

```
while (s[i] != '\0' && t[j] != '\0')
```



```
while (s[i] != '\0' || t[j] != '\0')
```



问题四——学习语言的目的→考法

◆ 层次一：会基本语法。

测试小卷（用法），每周1次，1小时，晚自习、学习部。

◆ 层次二：会读程序、会写程序。

基础编程训练100题，包括各大公司笔试中有关C语言试题。

◆ 层次三：会快速地写程序。

代码量，要求2000~5000有效代码，加入ACM实验室。

存在的问题：如何组织、如何考核→合理的学习系统

◆ 层次四：会用计算机解决（简单的）实际问题。

问题四——初学程序者要解决什么问题？

一个建议：不建议做（只做）XXXX管理系统——结构体数组的增、删、改、查、统。

原因一：数据结构中，线性表的顺序存储实现。

原因二：数组的基本操作，是最基本的训练。

原因三：所有XXXX管理系统大同小异。

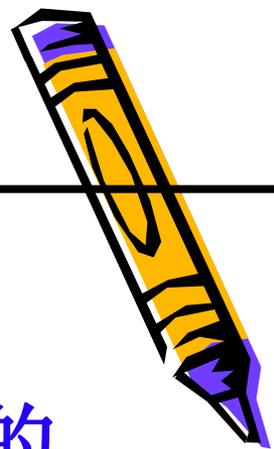
- ◆ **有趣**：数论、几何、物理、游戏。
- ◆ **实用**：各大公司笔试试题，用到各种语法知识。
- ◆ **提升**：基本的算法设计技术——蛮力法、递推法、分治法（递归）、贪心法等简单尝试。

一个观点

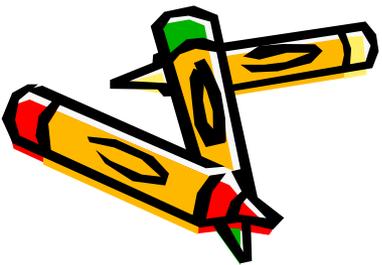
培育教学素养，提高教学水平——课程建设、教学研究和编写教材是载体。

- 教师不一定为了评精品课而进行课程建设
- 教师不一定为了获教学成果而进行教学研究
- 教师不一定为了出版教材而编写教材

干亮丽事业，度充实人生！



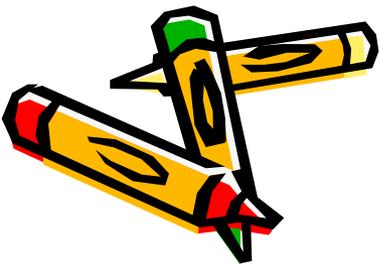
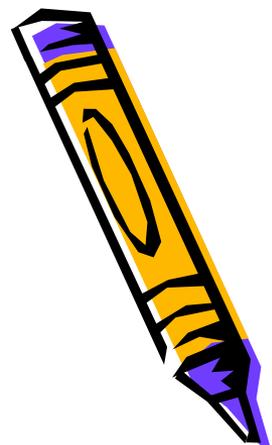
教师是人类最神圣的职业，教育是快乐的、幸福的，是充实的、满足的。祝愿各位老师
在课程建设和教学研究中形成更加鲜明的教学风格，提升教学智慧，提高教学素养。



谢谢!

敬请各位老师批评指正!

希望和各位老师交流沟通!



第5届全国高等学校计算机程序设计课程研讨会



扫一扫访问大会官网

<http://dbl原因lab.xmu.edu.cn/post/5120/>

