

## 2 理性认识计算机程序

郑海永

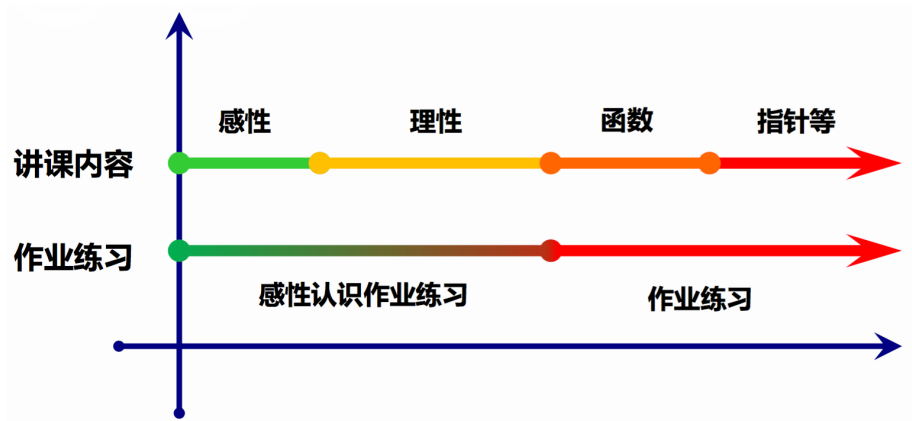
zhenghaiyong@gmail.com

<http://vision.ouc.edu.cn/~zhenghaiyong/courses/>

中国海洋大学 信息科学与工程学院 电子工程系



## 接下来的学习进度



# 目录 I

- ① C 语言的由来、标准和构成
  - C 语言的由来
  - C 语言的标准
  - C 语言的构成
- ② C 语言中的数据成分
- ③ C 语言中的运算成分
- ④ C 语言中的控制成分
- ⑤ C 程序中的数组
- ⑥ C 程序中的字符串

# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

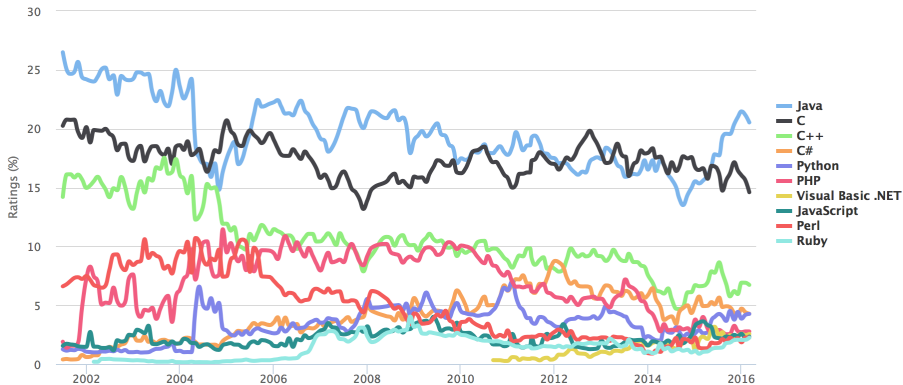
## 6 C 程序中的字符串



[http://www.tiobe.com/tiobe\\_index](http://www.tiobe.com/tiobe_index)

## TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)





# 程序员技术练级攻略<sup>1</sup>

## 前言

- 不要乱买书，不要乱追新技术新名词。
- 回顾一下历史，你才能明白明天会是什么样。
- 一定要动手，例子不管多简单，都建议至少自己手敲一遍。
- **一定要学会思考**，思考为什么要这样，而不是那样。
- **还要举一反三的思考！**

<sup>1</sup>来源：<http://CoolShell.cn>

# 程序员技术练级攻略<sup>1</sup>

## Unix/Linux VS. Windows

- 现在的用户界面主宰：Web 和移动设备 iOS 或 Android
- 低成本高性能 Linux 及各种开源技术架构胜过高成本 Windows
- 微软产品变化太快，很不持久～玩弄程序员？
- 趋势：**前端 Web+ 移动，后台 Linux+ 开源。**

<sup>1</sup>来源：<http://CoolShell.cn>

# 程序员技术练级攻略<sup>1</sup>

## 启蒙入门

- ❶ 学习一门脚本语言（如 Python/Ruby）
- ❷ 用熟一种程序员的编辑器（不是 IDE）和一些基本工具
- ❸ 熟悉 Unix/Linux Shell 和常见的命令行
- ❹ 学习 Web 基础（HTML/CSS/JS）+ 服务器端技术（LAMP）

---

<sup>1</sup>来源：<http://CoolShell.cn>

# 程序员技术练级攻略<sup>1</sup>

## 启蒙入门

- ❶ 学习一门脚本语言（如 Python/Ruby）
  - ▶ 处理文本文件，读一个本地文件，逐行处理。
  - ▶ 遍历本地文件系统（如写程序统计一个目录下所有文件大小并按各种条件排序）。
  - ▶ 跟数据库打交道（如统计数据库里条目数量）。
  - ▶ 学会用各种 `print` 简单粗暴方式进行调试
  - ▶ 学会用 Google
- ❷ 用熟一种程序员的编辑器（不是 IDE）和一些基本工具
- ❸ 熟悉 Unix/Linux Shell 和常见的命令行
- ❹ 学习 Web 基础（HTML/CSS/JS）+ 服务器端技术（LAMP）

<sup>1</sup>来源：<http://CoolShell.cn>

# 程序员技术练级攻略<sup>1</sup>

## 启蒙入门

- ❶ 学习一门脚本语言（如 Python/Ruby）
- ❷ 用熟一种程序员的编辑器（不是 IDE）和一些基本工具
  - ▶ Vim / Emacs / Notepad++，学会如何配置代码补全、外观、外部命令等。
  - ▶ Source Insight（或 ctag）
  - ▶ Not Only Cool! 更有效率！
- ❸ 熟悉 Unix/Linux Shell 和常见的命令行
- ❹ 学习 Web 基础（HTML/CSS/JS）+ 服务器端技术（LAMP）

---

<sup>1</sup>来源：<http://CoolShell.cn>

# 程序员技术练级攻略<sup>1</sup>

## 启蒙入门

- ❶ 学习一门脚本语言（如 Python/Ruby）
- ❷ 用熟一种程序员的编辑器（不是 IDE）和一些基本工具
- ❸ 熟悉 Unix/Linux Shell 和常见的命令行
  - ▶ 一定要少用少用图形界面
  - ▶ 学会使用 `man` 来查看帮助
  - ▶ 文件系统结构和基本操作
    - `ls chmod chown rm find ln cat mount mkdir tar gzip`
  - ▶ 学会使用一些文本操作命令
    - `sed awk grep tail less more`
  - ▶ 学会使用一些管理命令
    - `ps top lsof netstat kill tcpdump iptables dd`
  - ▶ 了解/etc 目录下的各种配置文件，学会查看/var/log 下的系统日志以及/proc 下的系统运行信息
  - ▶ 了解正则表达式，使用正则表达式来查找文件。
- ❹ 学习 Web 基础（HTML/CSS/JS）+ 服务器端技术（LAMP）

# 程序员技术练级攻略<sup>1</sup>

## 进阶加深

- ❶ C 语言和操作系统调用（指针、内存模型，数据结构与算法）
- ❷ 学习 Java
- ❸ Web 的安全与架构
- ❹ 学习关系型数据库（MySQL）
- ❺ 一些开发工具（Git 或 SVN、gdb、Makefile 等）

---

<sup>1</sup>来源：<http://CoolShell.cn>

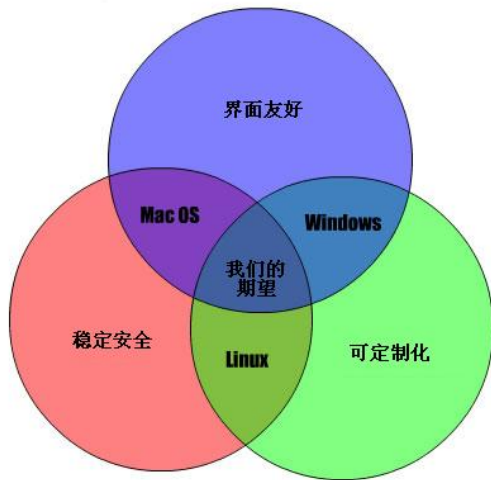
# 程序员技术练级攻略<sup>1</sup>

## 高级深入

- ① C++/Java 和面向对象
- ② 加强系统了解（Unix 编程、网络编程，TCP/IP 详解，等）
- ③ 系统架构（负载均衡，多层分布式系统，P2P 式，服务器备份，虚拟化，Hadoop，NoSQL，等）

---

<sup>1</sup>来源：<http://CoolShell.cn>



## 粉丝眼中的操作系统

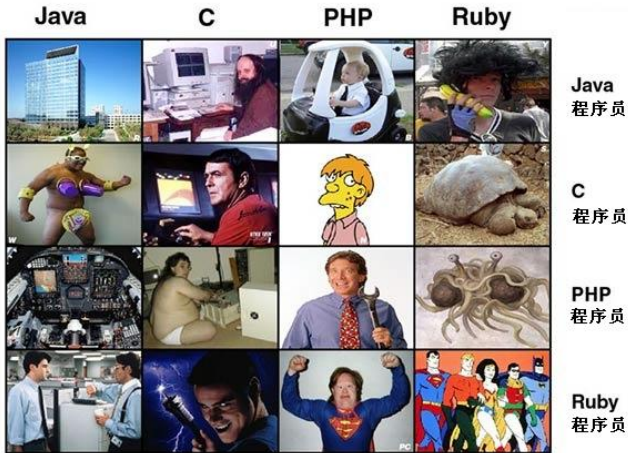


Mac  
粉丝

Windows  
粉丝

Linux  
粉丝

## 程序员眼中的编程语言



# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串

## 它为什么叫“C”语言？



# 程序设计语言的分类

## 低级语言之机器语言

|                        |                |
|------------------------|----------------|
| 0000000010000000001000 | 数据装入寄存器 0      |
| 000000001000100001010  | 数据装入寄存器 1      |
| 000001010000000000001  | 寄存器 0 与 1 的数据乘 |
| 000000001000100001100  | 数据装入寄存器 1      |
| 000001000000000000001  | 寄存器 0 与 1 的数据加 |
| 0000000100000000001110 | 保存寄存器 0 里的数据   |

# 程序设计语言的分类

## 低级语言之汇编语言

|          |                |
|----------|----------------|
| load 0 a | 数据装入寄存器 0      |
| load 1 b | 数据装入寄存器 1      |
| mult 0 1 | 寄存器 0 与 1 的数据乘 |
| load 1 c | 数据装入寄存器 1      |
| add 0 1  | 寄存器 0 与 1 的数据加 |
| save 0 d | 保存寄存器 0 里的数据   |

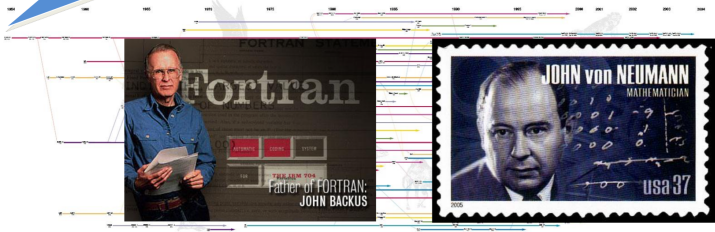
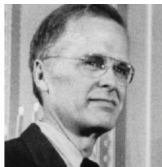
# 程序设计语言的分类

## 高级语言之 C

```
          d = a * b + c
0000000010000000001000  load 0 a
0000000010001000001010  load 1 b
0000010100000000000001  mult 0 1
0000000010001000001100  load 1 c
0000010000000000000001  add 0 1
0000001000000000001110  save 0 d
```

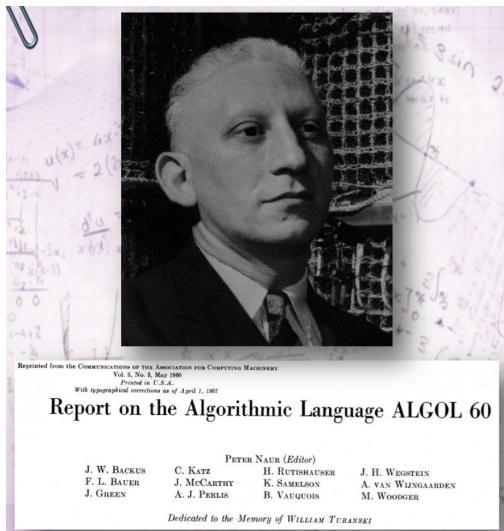
# 高级程序设计语言

1954 -1956年 IBM 的John Backus 和他的研究小组研发了FORTRAN( FORMula TRANslation)



[http://staff.pausd.org/~cbly/1web\\_design/12b\\_final/daniel/history.html](http://staff.pausd.org/~cbly/1web_design/12b_final/daniel/history.html)

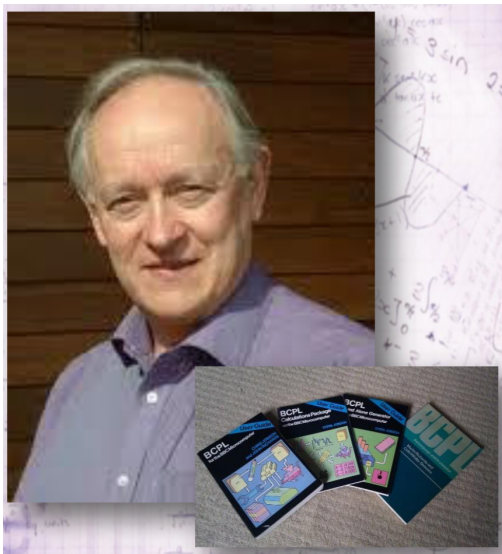
## C 程序设计语言的历史



1960 年 1 月

- Alan J. Perlis
- 软件专家讨论会（巴黎）
- 发表“算法语言 Algol 60 报告”
- 宣告程序设计语言 Algol 60 诞生
- 计算科学里程碑
- (A 语言)

## C 程序设计语言的历史



### 1963-1967 年

- BCPL 语言
- 1963 年，剑桥大学在 Algol 60 基础上推出 CPL (Combined Programming Language) 语言，但规模比较大，难以实现。
- 1967 年，剑桥大学 Martin Richards 对 CPL 语言作了简化，推出了 BCPL (Basic Combined Programming Language) 语言。

## C 程序设计语言的历史



### 1972-1973 年

- B 语言：贝尔实验室 Ken Thompson 设计出 B 语言，并用 B 语言写第一个 UNIX 操作系统，在 PDP-7 上实现。
- C 语言：1972-1973 年间，Dennis Ritchie 和 Ken Thompson 在 B 语言基础上发展和完善出 C 语言，并重写 UNIX。

# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串

# C 程序设计语言的版本

## K&R C

- 1978 年，Kernighan 和 Ritchie 的《The C Programming Language》第一版出版，一直被广泛作为 C 语言事实上的规范，称“**K&R C**”。

## ANSI C 和 ISO C

- 1989 年，C 语言被 ANSI 标准化，对 K&R C 进行了扩展，包括了一些新的特性，也规定了一套标准函数库。
- ISO 成立 WG14 工作组来规定国际标准的 C 语言；通过对 ANSI 标准的少量修改，最终通过了 **ISO 9899:1990**；随后 ISO 标准被 ANSI 采纳。

## C99

- 在 ANSI 标准化后，WG14 小组继续致力于改进 C 语言；新的标准很快推出：**ISO 9899:1999**；这个标准就是通常提及的 **C99**；它被 ANSI 于 2000 年 3 月采用。

## C11

- 2011 年 12 月 8 日，ISO 正式公布 C 语言新的国际标准草案：**ISO/IEC 9899:2011**，即 **C11**。
- [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail)。

# C 程序设计语言的版本

## K&R C

- 1978 年，Kernighan 和 Ritchie 的《The C Programming Language》第一版出版，一直被广泛作为 C 语言事实上的规范，称“**K&R C**”。

## ANSI C 和 ISO C

- 1989 年，C 语言被 ANSI 标准化，对 K&R C 进行了扩展，包括了一些新的特性，也规定了一套标准函数库。
- ISO 成立 WG14 工作组来规定国际标准的 C 语言；通过对 ANSI 标准的少量修改，最终通过了**ISO 9899:1990**；随后 ISO 标准被 ANSI 采纳。

## C99

- 在 ANSI 标准化后，WG14 小组继续致力于改进 C 语言；新的标准很快推出：**ISO 9899:1999**；这个标准就是通常提及的**C99**；它被 ANSI 于 2000 年 3 月采用。

## C11

- 2011 年 12 月 8 日，ISO 正式公布 C 语言新的国际标准草案：**ISO/IEC 9899:2011**，即**C11**。
- [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail)。

# C 程序设计语言的版本

## K&R C

- 1978 年，Kernighan 和 Ritchie 的《The C Programming Language》第一版出版，一直被广泛作为 C 语言事实上的规范，称“**K&R C**”。

## ANSI C 和 ISO C

- 1989 年，C 语言被 ANSI 标准化，对 K&R C 进行了扩展，包括了一些新的特性，也规定了一套标准函数库。
- ISO 成立 WG14 工作组来规定国际标准的 C 语言；通过对 ANSI 标准的少量修改，最终通过了**ISO 9899:1990**；随后 ISO 标准被 ANSI 采纳。

## C99

- 在 ANSI 标准化后，WG14 小组继续致力于改进 C 语言；新的标准很快推出：**ISO 9899:1999**；这个标准就是通常提及的**C99**；它被 ANSI 于 2000 年 3 月采用。

## C11

- 2011 年 12 月 8 日，ISO 正式公布 C 语言新的国际标准草案：**ISO/IEC 9899:2011**，即**C11**。
- [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail)。

## C 程序设计语言的版本

### K&R C

- 1978 年，Kernighan 和 Ritchie 的《The C Programming Language》第一版出版，一直被广泛作为 C 语言事实上的规范，称“**K&R C**”。

### ANSI C 和 ISO C

- 1989 年，C 语言被 ANSI 标准化，对 K&R C 进行了扩展，包括了一些新的特性，也规定了一套标准函数库。
- ISO 成立 WG14 工作组来规定国际标准的 C 语言；通过对 ANSI 标准的少量修改，最终通过了 **ISO 9899:1990**；随后 ISO 标准被 ANSI 采纳。

### C99

- 在 ANSI 标准化后，WG14 小组继续致力于改进 C 语言；新的标准很快推出：**ISO 9899:1999**；这个标准就是通常提及的 **C99**；它被 ANSI 于 2000 年 3 月采用。

### C11

- 2011 年 12 月 8 日，ISO 正式公布 C 语言新的国际标准草案：**ISO/IEC 9899:2011**，即 **C11**。
- [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail)。

## 注意

### C语言规范定义的非常宽泛：

- long 型数据长度不短于 int 型
- short 型不长于 int 型

### 经常导致：

- 相同的程序在不同编译器上具有不同解释
- 相同的程序在不同平台上运行结果不同
  - ▶ 例如：整型变量定义；对 ++、--的解释；输入输出赋值顺序的不同；浮点数计算精度的不同；……

## 注意

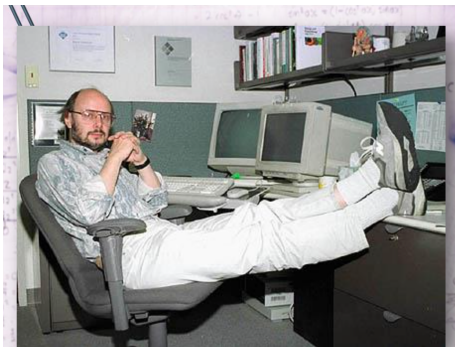
### C语言规范定义的非常宽泛：

- long 型数据长度不短于 int 型
- short 型不长于 int 型

### 经常导致：

- 相同的程序在不同编译器上具有不同解释
- 相同的程序在不同平台上运行结果不同
  - ▶ 例如：整型变量定义；对 ++、-- 的解释；输入输出赋值顺序的不同；浮点数计算精度的不同；……

# C++



**Bjarne Stroustrup**  
**本贾尼·斯特劳斯特卢普**

[http://www.iso.org/iso/home/store/catalogue\\_e\\_tc/catalogue\\_detail.htm?csnumber=50372](http://www.iso.org/iso/home/store/catalogue_e_tc/catalogue_detail.htm?csnumber=50372)

- 1979 年，贝尔实验室 Bjarne Stroustrup 开发了一种语言，被称为“C with Classes”，后来演化为“C++”。
- 1985 年 10 月，Bjarne 博士完成经典巨著《The C++ Programming Language》第一版。
- 1998 年 11 月 ISO 颁布了 C++ 程序设计语言的国际标准 ISO/IEC 14882:1998。
- ISO 于 2011 年 9 月 1 日发布了 ISO/IEC 14882:2011 即 C++ 2011。

# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串

## 我们学的是什么高级语言？

C 语言部分

C++ 语言  
(面向对象部分)

## 应该如何学习编程语言

# 程序设计语言的构成

语言的种类千差万别，但是，一般来说，基本成分不外四种：

数据成分 用以描述程序中所涉及的数据；

运算成分 用以描述程序中所包含的运算；

控制成分 用以表达程序中的控制构造；

传输成分 用以表达程序中数据的传输。

——计算机科学技术百科

# 如果让我来学习一门程序设计语言

## 数据成分

- 有哪些数据类型？如何使用？

## 运算成分

- 有哪些运算符号？如何使用？

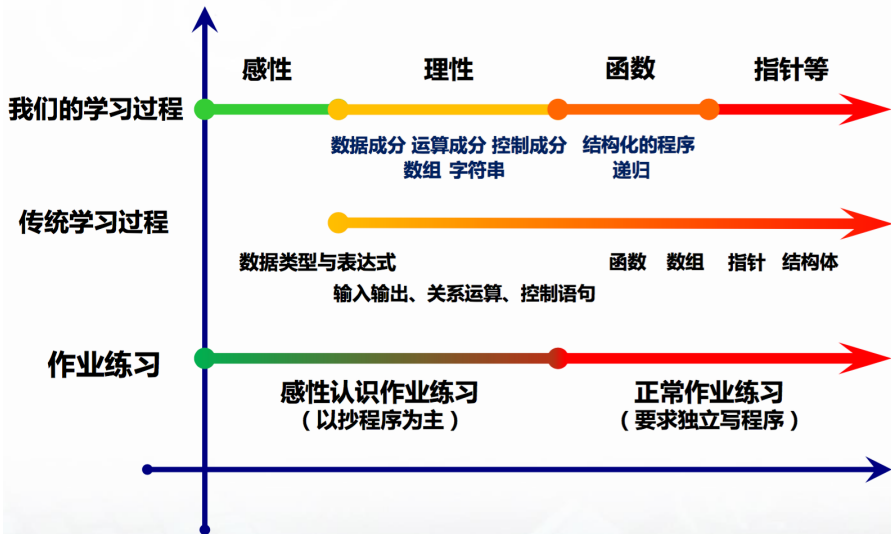
## 控制成分

- 三种类型的控制语句是如何写的？

## 传输成分

- 在程序中如何输入和输出数据？

```
1  #include<iostream>
2  using namespace std;
3  int main( )
4  {
5      int number[45] = {78, 56, 69, 31, 36, 67, 31, 47, 69, 34,
↪  45, 74, 61, 82, 43, 41, 76, 79, 81, 66, 54, 50, 76, 51,
↪  53, 28, 74, 39, 45, 61, 52, 41, 43, 75, 78, 84, 72, 51,
↪  43, 64, 75, 81, 69, 55, 74};
6      int max = 0;
7      int i = 0;
8      for(i = 0; i < 45; i++)
9      {
10         if(number[i] > max) max = number[i];
11     }
12     cout<<"The Maximal Number is: "<<max;
13     return 0;
14 }
```



# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串

# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串

# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串

# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串

# 内容提要 I

## 1 C 语言的由来、标准和构成

- C 语言的由来
- C 语言的标准
- C 语言的构成

## 2 C 语言中的数据成分

## 3 C 语言中的运算成分

## 4 C 语言中的控制成分

## 5 C 程序中的数组

## 6 C 程序中的字符串