



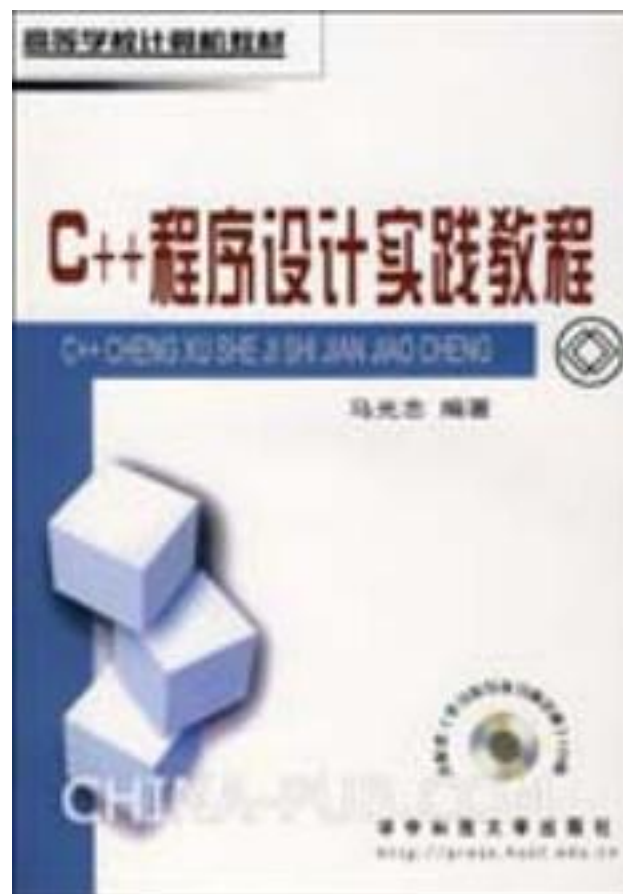
# C++程序设计精要教程

华中科技大学

# C++程序设计精要教程

马光志 编著

人民邮电出版社  
北京



老版教材：C++程序设计实践教程  
出版：华中科技大学出版社  
编著：马光志



# 第1章 C++引论

## ◆1.1 计算机的体系结构

**运算器**包括一组寄存器，用于运算

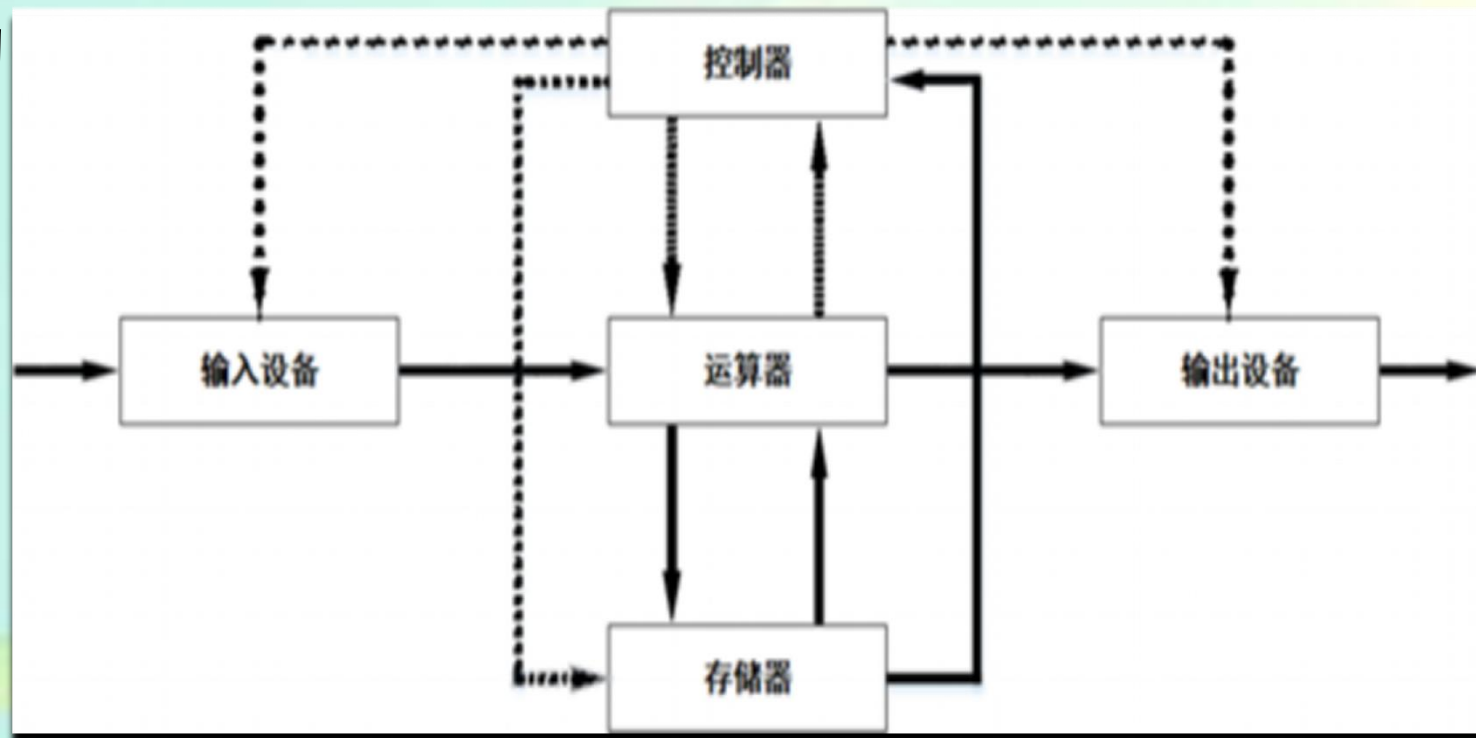
**控制器**控制指令流向及输入/输出设备

**存储器**用于存储二进制数据及指令

**输入设备**用于输入数据

**输出设备**用于输出数据

**寄存器**用于缓存数据，早期计算机用存储器代替。**存储器**称为内存，有RAM和ROM内存。有的设备既可以用作输入，也可以用作输出，例如磁盘设备；**控制台**由键盘及显示器构成。





# 第1章 C++引论

## 数据与程序的存储方式：

- 内存按字节编址，每个字节能存一个字符，一个整数占用多个字节。  
Windows分配内存的最小单位为节：1节=16字节。
- 一个字节由8位二进制构成，二进制位没有地址。
- 数据和指令均于二进制形式存于内存，数据可以当作指令看待，指令也可以当作数据看待，当指令计数器指向某个内存地址，自此地址开始的数据被当作指令执行。
- 从键盘输入的字符，用ASCII码表示，可转换后存入内存，从内存输出数据也经常需要转换成ASCII，再输出到显示器等设备。
- 字符'2'的ASCII码为50，与整数2的值不同，转换方法为 $2 = '2' - '0'$ 。
- 字母和数字的ASCII码是连续编码的；字母分大小写，ASCII码值不一样。

# 第1章 C++引论

## ◆1.2 进制及其转换和运算

- 采用二进制原因：电路成本接近最经济、更稳定，位逻辑运算更易实现，可通过补码将减法用加法运算实现。
- C++采用的进制：C++整数可采用八进制、十进制、十六进制表示。
- 二进制运算特点：二进制只有0、1两位数字；加法运算逢二进一；位运算包括移位、按位与、按位或、按位异或、按位求反；逻辑运算包括逻辑与、逻辑或、逻辑非等运算。
- 二、十进制转换：十进制数转换为二进制数时，通过连续除于2完成，余数为转后得到的二进制数字；二进制数转换为十进制数时，通过连续将二进制高位数值乘以2加上低一位数字值得到。
- 二、八进制转换：三位二进制数字可和一位八进制数字相互转换。
- 二、十六进制转换：四位二进制数字可和一位十六进制数字相互转换。

# 第1章 C++引论

## 二进制运算：

- 按位与&：对应二进制位都为1时结果为1，否则结果为0。
- 按位或|：对应二进制位都为0时结果为0，否则结果为1。
- 按位异或^：对应二进制位相同时结果为0，否则结果为1。

	10110010		10110010		10110010
按位与&	11010011	按位或	11010011	按位异或^	11010011
<hr/>					
	10010010		11110011		01100001

连续3个异或运算可交换两个变量x、y的值： $x=x^y$ ； $y=x^y$ ； $x=x^y$ ；

# 第1章 C++引论

## ◆1.3 80X86系列汇编语言

- C++可编译为汇编程序：C++函数编译形成代码段CS，C++全局和静态变量编译形成数据段DS，C++函数参数和局部自动变量编译后存于栈段SS。
- C++函数返回的int值存储在通用寄存器EAX中，返回的double值存储在EAX+EDX中，返回的对象值则实现起来很复杂。
- C++函数的参数通过压栈完成传递，调用完成后实参的值出栈，保持函数调用前后栈指针的值不变或平衡。函数局部自动变量通过在栈上内存实现。
- C++的变量名和函数名编译为汇编程序的变量名和函数名时，编译结果通常包含变量、函数（包括参数和返回值）的类型信息。
- 从C++程序编译得到的汇编程序在执行函数体前，通常会在栈上保存重要寄存器的值，返回前会从栈上出栈恢复这些重要寄存器的值，此类开销被称为调用开销。相对于编译得到的函数体汇编指令，调用开销有时较大。



# 第1章 C++引论

## 80X86系列汇编指令

- 汇编指令格式：操作码(位于低地址)—操作数(位于高地址)。注解用“;”。
- 汇编指令分类：数据传输如MOV，栈操作如PUSH和POP，数据运算如INC、AND、SUB，判断转移如CMP和JNZ，以及调用返回如CALL即RET等指令。
- 寄存器分类：段CS、DS、ES、SS，地址指针如ESI、EDI，ESP、EBP，数据如EAX、EBX、ECX、EDX，以及指令指针EIP和状态字如WSP等寄存器。
- 数据传输指令：MOV EAX, 0 ;将立即数0送至EAX寄存器  
MOV EBX, DOWRD PTR[EBP+8];
- 栈操作指令：PUSH EBP; ;保存EBP寄存器  
POP EBP; ;恢复EBP寄存器  
PUSH 4; ;将4压入栈中，常用于传递函数

参数



# 第1章 C++引论

## 80X86系列汇编指令

- 数据运算指令：  
DEC EAX ;将EAX的值减1送回EAX寄存器  
XOR EAX, EAX ;将EAX的值设置为0  
IMUL EAX, DOWRD PTR[EBP+8] ;
- 判断转移指令：  
CMP EAX, EBX ;比较EAX和EBX的值  
JG gthan ;如果EAX值>EBX值，则转至gthan执行
- 调用返回指令：  
CALL @@f\$qi ;调C++函数f(int)，@在汇编中类似字母  
RET ;返回到主调函数下一条语句执行

说明：调用前用PUSH指令传递实参，函数返回值通常存储在EAX中，返回后通过POP指令或“ADD ESP, …”将实参出栈，保持调用前后ESP的值不变(栈平衡)。参见例1.1及其汇编程序。VS2019调试时可用ctrl+f11组合键进入汇编程序。

# 第1章 C++引论



## ◆1.4 C++的历史及特点

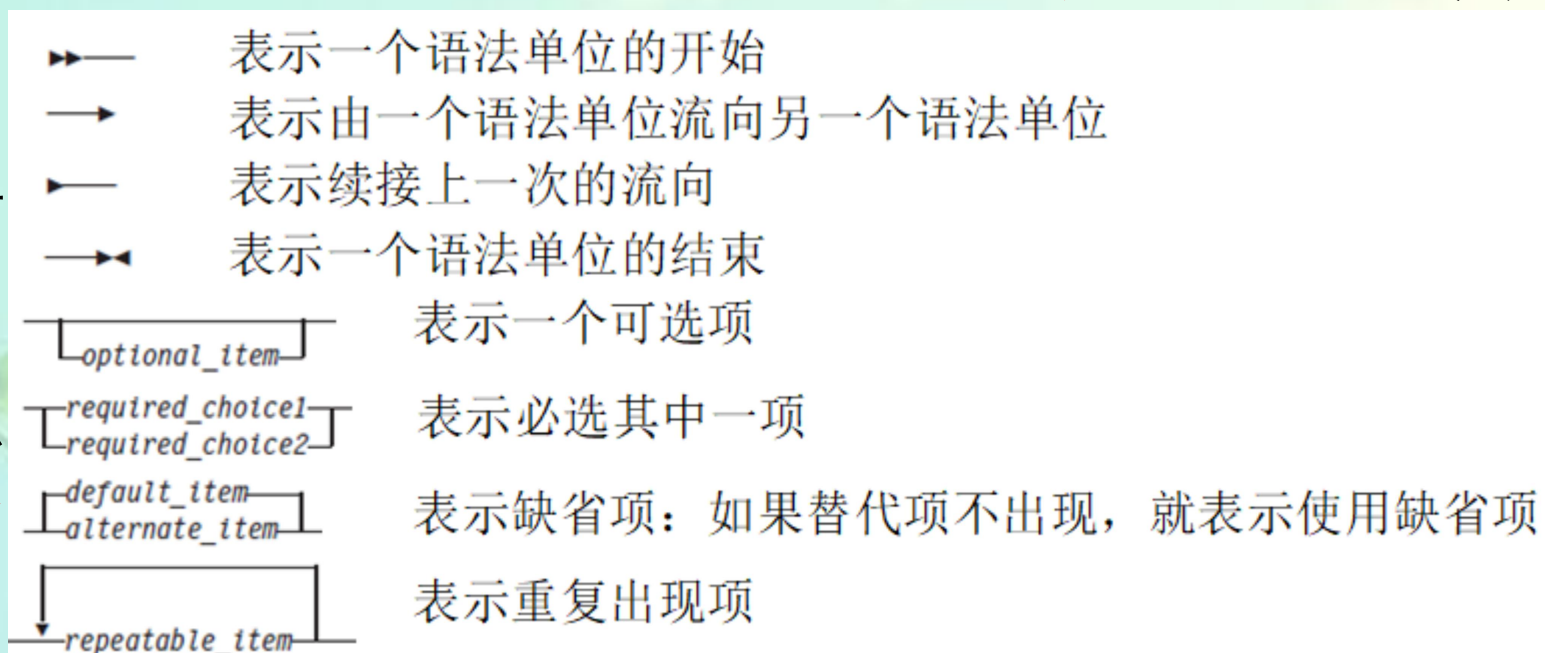
- C++之父：Bjarne Stroustrup(布贾尼·斯特劳斯特卢普)
- C++历史：BCPL(1967)→B(1969)→C(1971)→C with class(1978)→C++(1983)
- C++家族：Java、C#、JavaScript、Perl、PHP、Python、Ruby、D。
- C++特点：(1) C的超集；(2)强类型检查；(3)混合型面向对象；(4)支持多继承；(5)运算符重载；(6)支持移动语义；(7)异常及断言处理；(8)函数与类模板；(9)Lambda表达式；(10)支持类型推导；(11)支持名字空间。

总之：别的语言有什么特性，C++就希望有什么特性。因此，C++已经发展成为功能特性最为全面、最复杂的程序设计语言。

# 第1章 C++引论

## ◆1.5 语法图及程序流程图

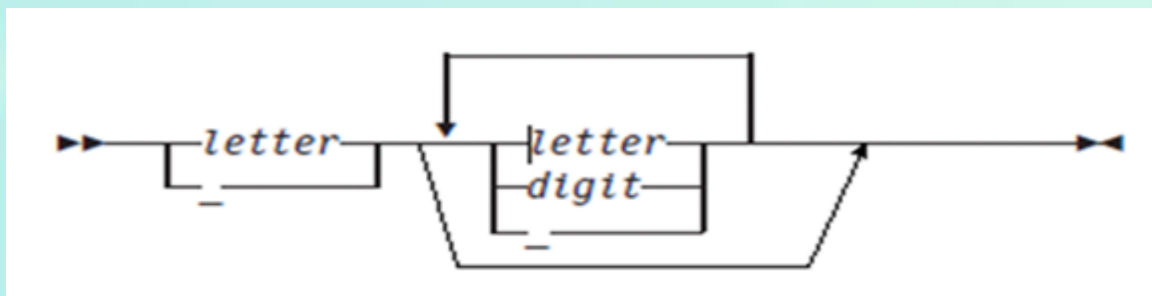
- 语法图是描述程序设计语言语法结构的图形语言。用倾斜文字表示一个概念。
- 语法图中的非倾斜文字表示：struct、if等**保留字**，=、+等**运算符**以及括号{ }等**分割符**等符号。
- 基于语法图的递归定义容易理解多重指针多重循环等概念。
- 例如：一个**标识符**是由**字母**或**下划线**开始的**字母**、**数字**和**下划线**的字符**序列**构成。





# 第1章 C++引论

## 标识符的语法图

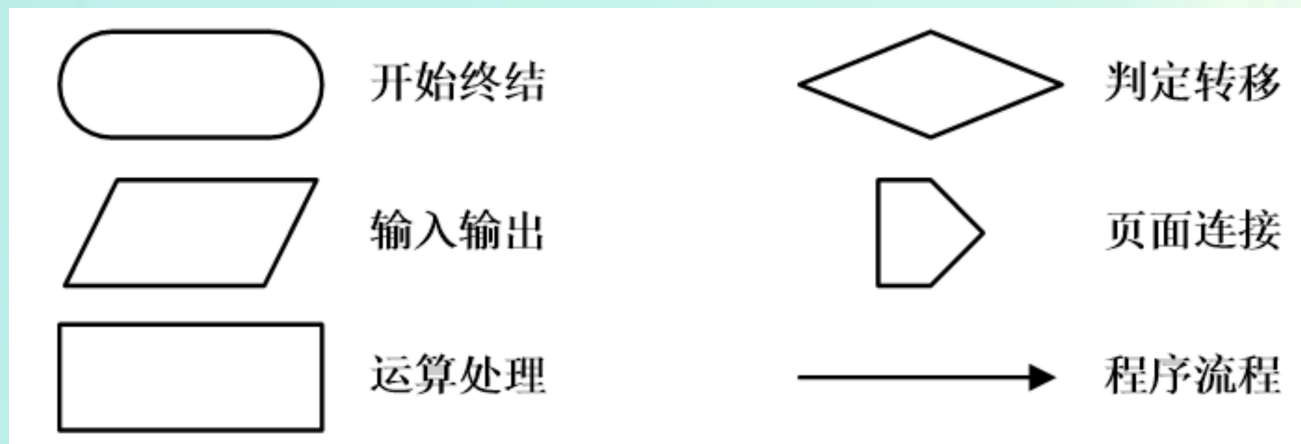


- 中 *letter* 和 *digit* 都是倾斜文字，这些概念可进一步用语法图定义
- 图中概念 *digit* 可用任意一个具体的值 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 进行替换。
- 合法的标识符：\_、a、A、\_1、\_aa123、abc 等。C++ 一个字符的标识符共有 53 个：\_，a...z，A...Z。C++ 的标识符区分大小写。

# 第1章 C++引论

## 程序流程图的画法

- 可使用微软的Visio画基本流程图，流程图的规范符号：



- 判定转移必须有2个分支出口，分别标有“yes”、“no”，或“是”、“否”。
- 注意事项：每个程序必须有且仅有1个开始和1个终结。参见图1.8。

# 第1章 C++引论

## ◆1.6 编译环境的安装与使用

- Microsoft Visual 2019能比较全面地支持C++2017国际标准。
- 具有较好的开发、调试、发布与维护功能。具有丰富的基础类库和专业类库。
- 不同编译器对C++2017的支持程度不同、不同操作系统执行结果可能不同。
- 建议采用64位电脑硬件、Windows 10、VS2019企业版，调试采用X86模式。
- 安装VS2019时，选择“使用C++的桌面开发”。
- 用“main”作为程序入口时，在“新建项目”窗口选“Windows控制台应用程序”。
- 调试时ctrl+f11可进入汇编指令。
- 鼠标停留在变量上可显示其值。
- 可单步、可设置条件断点调试。

```
#include "pch.h" //不要删除该行
#include <iostream>
int main()
{ std::cout << "Hello World!\n"; }
```