C与C++

C语言：

[C](https://baike.baidu.com/item/C/22337)语言是一门通用[计算机编程语言](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%BC%96%E7%A8%8B%E8%AF%AD%E8%A8%80" \t "_blank)，应用广泛。C语言的设计目标是提供一种能以简易的方式[编译](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91" \t "_blank)、处理低级[存储器](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%99%A8)、产生少量的[机器码](https://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E7%A0%81)以及不需要任何运行环境支持便能运行的编程语言。

尽管C语言提供了许多低级处理的功能，但仍然保持着良好跨平台的特性，以一个标准规格写出的C语言程序可在许多电脑平台上进行编译，甚至包含一些嵌入式[处理器](https://baike.baidu.com/item/%E5%A4%84%E7%90%86%E5%99%A8" \t "_blank)（单片机或称[MCU](https://baike.baidu.com/item/MCU/62773" \t "_blank)）以及超级电脑等作业平台。

它是把[高级语言](https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80)的基本结构和[语句](https://baike.baidu.com/item/%E8%AF%AD%E5%8F%A5)与低级语言的实用性结合起来的工作单元。

结构式语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及[调试](https://baike.baidu.com/item/%E8%B0%83%E8%AF%95)。C 语言是以函数形式提供给用户的，这些函数可方便的调用，并具有多种循环、条件语句控制程序流向，从而使程序完全[结构化](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96)。

由于标准的存在，使得几乎同样的C代码可用于多种操作系统，如Windows、[DOS](https://baike.baidu.com/item/DOS" \t "_blank)、[UNIX](https://baike.baidu.com/item/UNIX" \t "_blank)等等；也适用于多种机型。C语言对编写需要进行硬件操作的场合，优于其它高级语言。

可以直接进行靠近硬件的操作，但是C的指针操作不做保护，也给它带来了很多不安全的因素。[C++](https://baike.baidu.com/item/C%2B%2B)在这方面做了改进，在保留了指针操作的同时又增强了安全性，受到了一些用户的支持，但是，由于这些改进增加语言的复杂度，也为另一部分所诟病。Java则吸取了C++的教训，取消了[指针](https://baike.baidu.com/item/%E6%8C%87%E9%92%88)操作，也取消了C++改进中一些备受争议的地方，在安全性和适合性方面均取得良好的效果，但其本身解释在[虚拟机](https://baike.baidu.com/item/%E8%99%9A%E6%8B%9F%E6%9C%BA)中运行，运行效率低于C++/C。一般而言，C，C++，java被视为同一系的语言，它们长期占据着程序使用榜的前三名。

优点：

**1.简洁紧凑、灵活方便**

C语言一共只有32个[关键字](https://baike.baidu.com/item/%E5%85%B3%E9%94%AE%E5%AD%97" \t "_blank)，9种[控制语句](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E8%AF%AD%E5%8F%A5" \t "_blank)，程序书写形式自由，区分大小写。把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像[汇编语言](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96%E8%AF%AD%E8%A8%80/61826" \t "_blank)一样对位、[字节](https://baike.baidu.com/item/%E5%AD%97%E8%8A%82)和地址进行操作，而这三者是计算机最基本的工作单元。

**2.运算符丰富**

C语言的[运算符](https://baike.baidu.com/item/%E8%BF%90%E7%AE%97%E7%AC%A6" \t "_blank)包含的范围很广泛，共有34种运算符。C语言把[括号](https://baike.baidu.com/item/%E6%8B%AC%E5%8F%B7" \t "_blank)、[赋值](https://baike.baidu.com/item/%E8%B5%8B%E5%80%BC)、[强制类型转换](https://baike.baidu.com/item/%E5%BC%BA%E5%88%B6%E7%B1%BB%E5%9E%8B%E8%BD%AC%E6%8D%A2)等都作为运算符处理。从而使C语言的运算类型极其丰富，[表达式](https://baike.baidu.com/item/%E8%A1%A8%E8%BE%BE%E5%BC%8F" \t "_blank)类型多样化。灵活使用各种运算符可以实现在其它[高级语言](https://baike.baidu.com/item/%E9%AB%98%E7%BA%A7%E8%AF%AD%E8%A8%80)中难以实现的运算。

3、**数据类型丰富**

C语言的数据类型有：[整型](https://baike.baidu.com/item/%E6%95%B4%E5%9E%8B" \t "_blank)、实型、字符型、数组类型、指针类型、[结构体类型](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E4%BD%93%E7%B1%BB%E5%9E%8B)、共用体类型等。能用来实现各种复杂的数据结构的运算。并引入了指针概念，使程序效率更高。[13]

4、**表达方式灵活实用**

C语言提供多种运算符和表达式值的方法，对问题的表达可通过多种途径获得，其程序设计更主动、灵活。它语法限制不太严格，程序设计自由度大，如对整型量与字符型数据及逻辑型数据可以通用等。[14]

5、**允许直接访问物理地址，对硬件进行操作**

由于C语言允许直接访问[物理地址](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%9C%B0%E5%9D%80" \t "_blank)，可以直接对硬件进行操作，因此它既具有高级语言的功能，又具有低级语言的许多功能，能够像[汇编语言](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96%E8%AF%AD%E8%A8%80)一样对位（bit）、字节和地址进行操作，而这三者是计算机最基本的工作单元，可用来写[系统软件](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E8%BD%AF%E4%BB%B6)。

6、**生成目标代码质量高，程序执行效率高**

C语言描述问题比汇编语言迅速，工作量小、可读性好，易于[调试](https://baike.baidu.com/item/%E8%B0%83%E8%AF%95" \t "_blank)、修改和移植，而代码质量与汇编语言相当。C语言一般只比汇编程序生成的目标代码效率低10%～20%。

7、**可移植性好**

C语言在不同机器上的C[编译程序](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E7%A8%8B%E5%BA%8F)，86%的代码是公共的，所以C语言的编译程序便于移植。在一个环境上用C语言编写的程序，不改动或稍加改动，就可移植到另一个完全不同的环境中运行。

8、**表达力强**

C语言有丰富的数据结构和[运算符](https://baike.baidu.com/item/%E8%BF%90%E7%AE%97%E7%AC%A6)。包含了各种数据结构，如[整型](https://baike.baidu.com/item/%E6%95%B4%E5%9E%8B)、数组类型、指针类型和联合类型等，用来实现各种数据结构的运算。C语言的运算符有34种，范围很宽，灵活使用各种运算符可以实现难度极大的运算。

C语言能直接访问硬件的物理地址，能进行位（[bit](https://baike.baidu.com/item/bit)）操作。兼有高级语言和低级语言的许多优点。

它既可用来编写系统软件，又可用来开发应用软件，已成为一种通用程序设计语言。

另外C语言具有强大的图形功能，支持多种显示器和[驱动器](https://baike.baidu.com/item/%E9%A9%B1%E5%8A%A8%E5%99%A8" \t "_blank)。且计算功能、[逻辑判断](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E5%88%A4%E6%96%AD)功能强大。

**缺点**

1、 C语言的缺点主要表现在数据的封装性上，这一点使得C在数据的安全性上有很大缺陷，这也是C和C++的一大区别。

2、 C语言的语法限制不太严格，对变量的类型约束不严格，影响程序的安全性，对[数组下标越界](https://baike.baidu.com/item/%E6%95%B0%E7%BB%84%E4%B8%8B%E6%A0%87%E8%B6%8A%E7%95%8C" \t "_blank)不作检查等。从应用的角度，C语言比其他高级语言较难掌握。也就是说，对用C语言的人，要求对程序设计更熟练一些。

C++：

C++是C语言的继承，它既可以进行C语言的过程化程序设计，又可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行以继承和多态为特点的面向对象的程序设计。C++擅长面向对象程序设计的同时，还可以进行基于过程的程序设计，因而C++就适应的问题规模而论，大小由之。

C++不仅拥有计算机高效运行的实用性特征，同时还致力于提高大规模程序的编程质量与程序设计语言的问题描述能力。

C++语言的程序因为要体现高性能，所以都是编译型的。但其开发环境，为了方便测试，将调试环境做成解释型的。即开发过程中，以解释型的逐条语句执行方式来进行调试，以编译型的脱离开发环境而启动运行的方式来生成程序最终的执行代码。

生成程序是指将源码（*C++语句*）转换成一个可以运行的应用程序的过程。如果程序的编写是正确的，那么通常只需按一个功能键，即可搞定这个过程。该过程实际上分成两个步骤。

第一步是对程序进行编译，这需要用到编译器（*compiler*）。编译器将C++语句转换成机器码(也称为目标码)；如果这个步骤成功，下一步就是对程序进行链接，这需要用到链接器（*linker*）。链接器将编译获得机器码与C++库中的代码进行合并。C++库包含了执行某些常见任务的函数（*“函数”是子程序的另一种称呼*）。例如，一个C++库中包含标准的平方根函数sqrt，所以不必亲自计算平方根。C++库中还包含一些子程序，它们把数据发送到显示器，并知道如何读写硬盘上的数据文件。

C和C++:

C++是在C语言的基础上开发的一种面向对象编程语言，应用非常广泛。常用于系统开发，引擎开发等应用领域，支持类、封装、继承、多态等特性。C++语言灵活，运算符的数据结构丰富、具有结构化控制语句、程序执行效率高，而且同时具有高级语言与汇编语言的优点。[25]

C++语言是对C语言的扩充，从Simula中吸取了类，从ALGOL语言中吸取了运算符的一名多用、引用和在分程序中任何位置均可说明变量，综合了Ada语言的类属和Clu语言的模块特点，形成了抽象类，从Ada Clu和ML等语言吸取了异常处理，从BCPL语言中吸取了用//表示注释C++语言保持了C语言的紧凑灵活、高效以及易于移植性强等优点，它对数据抽象的支持主要在于类概念和机制，对面向对象风范的支持主要通过虚拟机制函数因C++语言既有数据抽象和面向对象能力，运行性能高，加上C语言的普及，而从C语言到C++语言的过渡较为平滑，以及C++语言与C语言的兼容程度可使数量巨大的C语言程序能方便地在C++语言环境中复用，使C++语言在短短几年内能流行。

（以下摘自CSDN）

C++是C的超集，也可以说C是C++的子集，因为C先出现。按常理说，C++编译器能够编译任何C程序，但是C和C++还是有一些小差别。

      例如C++增加了C不具有的关键字。这些关键字能作为函数和变量的标识符在C程序中使用，尽管C++包含了所有的C，但显然没有任何C++编译器能编译这样的C程序。

**C程序员可以省略函数原型，而C++不可以，一个不带参数的C函数原型必须把void写出来。而C++可以使用空参数列表**。

     C++中**new和delete是对内存分配的运算符**，取代了C中的**malloc**和**free**。

     标准C++中的字符串类取代了C标准C函数库头文件中的字符数组处理函数（C中没有字符串类型）。

     C++中用来做控制态输入输出的**iostream类库**替代了标准C中的**stdio函数库**。

     C++中的try/catch/throw异常处理机制取代了标准C中的setjmp()和longjmp()函数。

C++相对与C增加了一些关键字，如下：

typename bool dynamic\_cast mutable namespace

static\_cast using catch explicit new

virtual operator false private template

volatile const protected this wchar\_t

const\_cast public throw friend true

reinterpret\_cast try

bitor xor\_e and\_eq compl or\_eq

not\_eq bitand

在C++中还增加了**bool型变量**和**wchar\_t型变量**：

**布尔型变量是有两种逻辑状态的变量，它包含两个值：真和假**。如果在表达式中使用了布尔型变量，那么将根据变量值的真假而赋予整型值1或0。要把一个整型变量转换成布尔型变量，如果整型值为0，则其布尔型值为假；反之如果整型值为非0，则其布尔型值为真。布儿型变量在运行时通常用做标志，比如进行逻辑测试以改变程序流程。

C++相对与C增加了一些关键字，如下：

typename bool dynamic\_cast mutable namespace

static\_cast using catch explicit new

virtual operator false private template

volatile const protected this wchar\_t

const\_cast public throw friend true

reinterpret\_cast try

bitor xor\_e and\_eq compl or\_eq

not\_eq bitand

在C++中还增加了**bool型变量**和**wchar\_t型变量**：

**布尔型变量是有两种逻辑状态的变量，它包含两个值：真和假**。如果在表达式中使用了布尔型变量，那么将根据变量值的真假而赋予整型值1或0。要把一个整型变量转换成布尔型变量，如果整型值为0，则其布尔型值为假；反之如果整型值为非0，则其布尔型值为真。布儿型变量在运行时通常用做标志，比如进行逻辑测试以改变程序流程。

有时候，根据表达式的需要，某个数据需要被当成另外的数据类型来处理，这时，就需要强制编译器把变量或常数由声明时的类型转换成需要的类型。为此，就要使用强制类型转换说明，格式如下：

int\* iptr=(int\*) &table;

表达式的前缀(int\*)就是传统C风格的强制类型转换说明(typecast)，又可称为强制转换说明(cast)。强制转换说明告诉编译器把表达式转换成指定的类型。有些情况下强制转换是禁用的，例如不能把一个结构类型转换成其他任何类型。数字类型和数字类型、指针和指针之间可以相互转换。当然，数字类型和指针类型也可以相互转换，但通常认为这样做是不安全而且也是没必要的。强制类型转换可以避免编译器的警告。

在C语言中，输入输出是使用语句**scanf()**和**printf()**来实现的，而C++中是使用类来实现的。

**cin,cout,endl对象**，他们本身并不是C++语言的组成部分。虽然他们已经是ANSI标准C++中被定义，但是他们不是语言的内在组成部分。在C++中不提供内在的输入输出运算符，这与其他语言是不同的。输入和输出是通过C++类来实现的，cin和cout是这些类的实例，他们是在C++语言的外部实现。

     在C++语言中，有了一种新的注释方法，就是‘//’，在该行//后的所有说明都被编译器认为是注释，这种注释不能换行。C++中仍然保留了传统C语言的注释风格/\*……\*/。

声明函数时可以包含一个或多个用不到的形式参数。这种情况多出现在用一个通用的函数指针调用多个函数的场合，其中有些函数不需要函数指针声明中的所有参数。

尽管这样的用法是正确的，但大多数C和C++的编译器都会给出一个警告，说参数y在程序中没有被用到。为了避免这样的警告，C++允许声明一个无名形参，以**告诉编译器存在该参数，且调用者需要为其传递一个实际参数，但是函数不会用到这个参数**。

**C++函数的原型中可以声明一个或多个带有默认值的参数**。如果调用函数时，省略了相应的实际参数，那么编译器就会把默认值作为实际参数。

在C++中，**允许有相同的函数名，不过它们的参数类型不能完全相同**，这样这些函数就可以相互区别开来。而这在C语言中是不允许的。

C++语言中，允许变量定义语句在程序中的任何地方，只要在是使用它之前就可以；而C语言中，必须要在函数开头部分。而且C++允许重复定义变量，C语言也是做不到这一点的。

在C++语言中，仍然支持malloc()和free()来分配和释放内存，同时增加了new和delete来管理内存。

在C++中，**引用是一个经常使用的概念**。引用型变量是其他变量的一个别名，我们可以认为他们只是名字不相同，其他都是相同的。

C++中的引用是其他变量的别名。声明一个引用型变量，需要给他一个初始化值，在变量的生存周期内，该值不会改变。

C++中的引用是其他变量的别名。声明一个引用型变量，需要给他一个初始化值，在变量的生存周期内，该值不会改变。& 运算符定义了一个引用型变量：

int a;

int& b=a;

先声明一个名为a的变量，它还有一个别名b。我们可以认为是一个人，有一个真名，一个外号，以后不管是喊他a还是b，都是叫他这个人。同样，作为变量，以后对这两个标识符操作都会产生相同的效果。

和指针不同，引用变量的值不可改变。引用作为真实对象的别名，必须进行初始化，除非满足下列条件之一：

(1) 引用变量被声明为外部的，它可以在任何地方初始化

(2) 引用变量作为类的成员，在构造函数里对它进行初始化

(3) 引用变量作为函数声明的形参，在函数调用时，用调用者的实参来进行初始化

从表面上看，这两个函数没有明显区别，不过他们所花的时间却有很大差异，func2()函数所用的时间开销会比func2()函数少很多。它们还有一个差别，如果程序递归func1()，随着递归的深入，会因为栈的耗尽而崩溃，但func2()没有这样的担忧。

引用常常被用作函数的形参。以引用代替拷贝作为形参的优点：

引用避免了传递大型数据结构带来的额外开销

引用无须象指针那样需要使用\*和->等运算符

当函数把引用作为参数传递给另一个函数时，被调用函数将直接对参数在调用者中的拷贝进行操作，而不是产生一个局部的拷贝(传递变量本身是这样的)。这就称为以引用方式调用。把参数的值传递到被调用函数内部的拷贝中则称为以传值方式调用。