

C 中const用法浅析计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_C___E4_B8_ADcons_c97_646052.htm 看到const 关键字，很多程序员想到的可能是const 常量，这可有点象踩到陷井上还不知道自己危险了。读读以下文字会使你对c 中的const有一个全面的认识。const 是C 中常用的类型修饰符，有某些微妙的应用场合，如果没有弄清本源，则错误在所难免。本篇中将对const进行辨析。溯其本源，究其实质，希望能对大家理解const有所帮助，根据思维的承接关系，分为如下几个部分进行阐述。C 中为什么会引入const C 的提出者当初是基于什么样的目的引入（或者说保留）const关键字呢？，这是一个有趣又有益的话题，对理解const很有帮助。

1. 大家知道，C 有一个类型严格的编译系统，这使得C 程序的错误在编译阶段即可发现许多，从而使得出错率大为减少，因此，也成为了C 与C++相比，有着突出优点的一个方面。
2. C中很常见的预处理指令 `#define VariableName VariableValue` 可以很方便地进行值替代，这种值替代至少在三个方面优点突出：一是避免了意义模糊的数字出现，使得程序语义流畅清晰，如下例：`#define USER_NUM_MAX 107` 这样就避免了直接使用107带来的困惑。二是可以很方便地进行参数的调整与修改，如上例，当人数由107变为201时，改动此处即可，三是提高了程序的执行效率，由于使用了预编译器进行值替代，并不需要为这些常量分配存储空间，所以执行的效率较高。鉴于以上的优点，这种预定义指令的使用在程序中随处可见。
3. 说到这里，大家可能会迷惑上述的1点、2点与const有什么关系呢？好，

请接着向下看来：预处理语句虽然有以上的许多优点，但它有个比较致命的缺点，即，预处理语句仅仅只是简单值替代，缺乏类型的检测机制。这样预处理语句就不能享受C严格类型检查的好处，从而可能成为引发一系列错误的隐患。4

好了，第一阶段结论出来了：结论：Const推出的初始目的，正是为了取代预编译指令，消除它的缺点，同时继承它的优点。现在它的形式变成了：Const DataType VariableName = VariableValue . 为什么const能很好地取代预定义语句？const到底有什么大神通，使它可以振臂一挥取代预定义语句呢？

1. 首先，以const修饰的常量值，具有不可变性，这是它能取代预定义语句的基础。
2. 第二，很明显，它也同样可以避免意义模糊的数字出现，同样可以很方便地进行参数的调整和修改。
3. 第三，C的编译器通常不为普通const常量分配存储空间，而是将它们保存在符号表中，这使得它成为一个编译期间的常量，没有了存储与读内存的操作，使得它的效率也很高，同时，这也是它取代预定义语句的重要基础。这里，我要提一下，为什么说这一点是也是它能取代预定义语句的基础，这是因为，编译器不会去读存储的内容，如果编译器为const分配了存储空间，它就不能够成为一个编译期间的常量了。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com