初学者应当如何学习C 以及编程计算机二级考试 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__E5_88_9D_E5_AD_A6_E8_80_85_E5_c97_646350.htm Javascript是世界上最受误解的语言，其实C 何尝不是。坊间流传的错误的C 学习方法一抓就是一大把。我自己在学习C 的过程中也走了许多弯路，浪费了不少时间。 为什么会存在这么多错误认识?原因主要有三个，一是C 语言的细节太多。二是一些著名的C 书籍总在(不管有意还是无意)暗示语言细节的重要性和有趣。三是现代C 库的开发哲学必须用到一些犄角旮旯的语言细节(但注意，是库设计，不是日常编程)。这些共同塑造了C 社群的整体心态和哲学。 单是第一条还未必能够成气候，其它语言的细节也不少(尽管比起C 起来还是小巫见大巫)，就拿Javascript来说，作用域规则，名字查找，closure，for/in，这些都是细节，而且其中还有违反直觉的。但许多动态语言的程序员的理念我猜大约是学到哪用到哪罢。但C 就不一样了，学C 之人有一种类似于被暗示的潜在心态，就是一定要先把语言核心基本上吃透了才能下手写出漂亮的程序。这首先就错了。这个意识形成的原因在第二点，C 书籍。市面上的C 书籍不计其数，但有一个共同的缺点，就是讲语言细节的书太多《C gotchas》，《Effective C 》，《More Effective C 》，但无可厚非的是，C 是这样一门语言：要拿它满足现代编程理念的需求，尤其是C 库开发的需求，还必须得关注语言细节，乃至于在C 中利用语言细节已经成了一门学问。比如C 模板在设计之初根本没有想到模板元编程这回事，更没想到C 模板系统是图灵完备的，这也就导致了《Modern C

Design》和《C Template Metaprogramming》的惊世骇俗。 这些技术的出现为什么惊世骇俗，打个比方，就好比是一块大家都认为已经熟悉无比，再无秘密可言的土地上，突然某天有人挖到原来地下还蕴藏着最丰富的石油。在这之前的C 虽然也有一些细节，但也还算容易掌握，那可是C 程序员们的happy old times，因为C 的一切都一览无余，everything is figured out。然而《Modern C Design》的出世告诉人们，“ 瞧，还有多少细节你们没有掌握啊。 ”于是C 程序员们久违的激情被重燃起来，奋不顾身的踏入细节的沼泽中。尤其是，模板编程将C 的细节进一步挖掘到了极致我们干嘛关心涉及类对象的隐式转换的优先级高低?看看boost::is_base_of就可以知道有多诡异了。 但最大的问题还在于，对于这些细节的关注还真有它合适的理由：我们要开发现代模板库，要开发active library，就必须动用模板编程技术，要动用模板编程技术，就必须利用语言的犄角旮旯，enable_if，type_traits，甚至连早就古井无波的C宏也在乱世中重生，看看boost::preprocessor有多诡异就知道了，连C宏的图灵完备性(预编译期的)都被挖掘出来了。为什么要做这些?好玩?标榜?都不是，开发库的实际需求。但这也正是最大的悲哀了。在boost里面因实际需求而动用语言细节最终居然能神奇的完成任务的最好教材就是boost::foreach，这个小设施对语言细节的发掘达到了惊天地泣鬼神的地步，不信你先试着自己去看看它的源代码，再看看作者介绍它的文章吧。而boost::typeof也不甘其后C 语言里面有太多被“ 发现 ”而不是被“ 发明 ”的技术。难道最初无意设置这些语言规则的家伙们都是Oracles? 因为没有variadic templates，人们用宏加上缺省模板

参数来实现类似效果。因为没有concepts，人们用模板加上析构函数的细节来完成类似工作。因为没有typeof，人们用模板元编程和宏加上无尽的细节来实现目标… C 开发者们的DIY精神不可谓不强。 然而，如果仅仅是因为要开发优秀的库，那么涉及这些细节都还是情有可原的，至少在C 09出现并且编译器厂商跟上之前，这些都还能说是不得已而为之。但我们广大的C 程序员呢?大众是容易被误导的，我也曾经是。以为掌握了更多的语言细节就更牛，但实际却是那些语言细节十有八九是平时编程用都用不到的。C 中众多的细节虽然在库设计者手里面有其用武之地，但普通程序员则根本无需过多关注，尤其是没有实际动机的关注。一般性的编码实践准则，以及基本的编程能力和基本功，乃至基本的程序设计理论以及算法设计。才是真正需要花时间掌握的东西。 学习最佳编码实践比学习C 更重要。看优秀的代码也比埋头用差劲的编码方式写垃圾代码要有效。直接、清晰、明了、KISS地表达意图比玩编码花招要重要… 避免去过问任何语言细节，除非必要。这个必要是指在实际编程当中遇到问题，这样就算需要过问细节，也是最省事的，懒惰者原则嘛。一个掌握了基本的编程理念并有较强学习能力的程序员在用一门陌生的语言编程时就算拿着那本语言的圣经从索引翻起也可以编出合格的程序来。十年学会编程不是指对每门语言都得十年，那一辈子才能学几门语言哪，如果按字母顺序学的话一辈子都别指望学到Ruby了.十年学习编程更不是指先把语言特性从粗到细全都吃透才敢下手编程，在实践中提高才是最重要的。 至于这种抠语言细节的哲学为何能在社群里面呈野火燎原之势，就是一个心理学的问题了。想像人们在论坛上讨论

问题时，一个对语言把握很细致的人肯定能够得到更多的佩服，而由于论坛上的问题大多是小问题，所以解决实际问题的真正能力并不能得到显现，也就是说，知识型的人能够得到更多佩服，后者便成为动力和仿效的砝码。然而真正的编程能力是与语言细节没关系的，熟练运用一门语言能够帮你最佳表达你的意图，但熟练运用一门语言绝不意味着要把它的边边角角全都记住。懂得一些常识，有了编程的基本直觉，遇到一些细节错误的时候再去查书，是最节省时间的办法。C 的书，Bjarne的圣经《The C Programming Language》是高屋建瓴的。《大规模C 程序设计》是挺务实的。《Accelerated C 》是最佳入门的。《C Templates》是仅作参考的。《C Template Metaprogramming》是精力过剩者可以玩一玩的，普通程序员碰都别碰的。《ISO.IEC C Standard 14882》不是拿来读的。Bjarne最近在做C 的教育，新书是绝对可以期待的。
P.S. 关于如何学习编程，g9的blog上有许多精彩的文章：这里，这里，这里，这里… 实际上，我建议你去把g9老大的blog翻个底朝天 :P 再P.S. 书单?我是遑于给出一个类似《C 初学者必读》这种书单的。C 的书不计其数，被公认的好书也不胜枚举。只不过有些书容易给初学者造成一种错觉，就是“ 学习C 就应该是这个样子的 ”。比如有朋友提到的《高质量C/C 编程》，这本书有价值，但不适合初学者，初学者读这样的书容易一叶障目不见泰山。实际上，正确的态度是，细节是必要的。但细节是次要的。其实学习编程我觉得应该最先学习如何用伪码表达思想呢，君不见《Introduction to Algorithm》里面的代码?《TAOCP》中的代码?哦，对了它们是自己建立的语言，但这种仅教学目的的语言的目的就是为了避免让

写程序的人一开始就忘了写程序是为了完成功能，以为写程序就是和语言细节作斗争了。Bjarne说程序的正确性最重要，boost的编码标准里面也将正确性列在性能前面。 此外，一旦建立了正确的学习编程的理念，其实什么书(只要不是太垃圾的)都有些用处。都当成参考书，用的时候从目录或索引翻，基本就对了。 再再P.S. myan老大和g9老大都给出了许多精彩的见解。我不得不再加上一个P.S。具体我就不摘录了，如果你读到这里，请务必往下看他们的评论。转载者别忘了转载他们的评论:-) 许多朋友都问我同一个问题，到底要不要学习C 。其实这个问题问得很没有意义。“ 学C ”和“ 不学C ”这个二分法是没意义的，为什么?因为这个问题很表面，甚至很浮躁。重要的不是你掌握的语言，而是你掌握的能力，借用myan老大的话，“ 重要的是这个磨练过程，而不是结果，要的是你粗壮的腿，而不是你身上背的那袋盐巴。 ”。此外学习C 的意义其实真的是醉翁之意不在酒，像C/C 这种系统级语言，在学习的过程中必须要涉及到一些底层知识，如内存管理、编译连接系统、汇编语言、硬件体系结构等等等等知识(注意，这不包括过分犄角旮旯的语言枝节)。这些东西也就是所谓的内功了(其实最最重要的内功还是长期学习所磨练出来的自学能力)。对此大嘴Joel在《Joel On Software》里面提到的漏洞抽象定律阐述得就非常漂亮。 所以，答案是，让你成为高手的并不是你掌握什么语言，精通C 未必就能让你成为高手，不精通C 也未必就能让你成为低手。我想大家都不会怀疑g9老大如果要抄起C 做一个项目的话会比大多数自认熟练C 的人要做得漂亮。所以关键的不是语言这个表层的东西，而是底下的本质矛盾。当然，不是说那就什么语言

都不要学了，按照一种曹操的逻辑，“ 天下语言，唯imperative与declarative耳 ”。C 是前者里面最复杂的一种，支持最广泛的编程范式。借用当初数学系入学大会上一个老师的话，“ 你数学都学了，还有什么不能学的呢? ”。学语言是一个途径，如果你把它用来磨练自己，可以。如果你把它用来作为学习系统底层知识的钥匙，可以。如果你把它用来作为学习如何编写优秀的代码，如何组织大型的程序，如何进行抽象设计，可以。如果掉书袋，光啃细节，我认为不可以(除非你必须要用到细节，像boost库的coder们)。 编辑推荐：2010年计算机等级考试二级C 冲刺全真模拟试题及答案汇总 2010年全国计算机等级考试二级C 模拟试题及答案汇总 全国计算机等级考试二级笔试样卷C 语言程序设计 全国计算机二级考试C 试题练习汇总 全国计算机等级考试二级C 语言程序设计标准预测试卷汇总 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