

初学编程的人们看过来 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/224/2021_2022__E5_88_9D_E5_AD_A6_E7_BC_96_E7_c100_224971.htm 我始终认为，对一个初学者来说，IT界的技术风潮是不可以追赶的，而且也没有能力去追赶。我时常看见自己的DDMM们把课本扔了，去卖些价格不菲的诸如C#, VB.Net 这样的大部头，这让我感到非常痛心。而许多搞不清指针是咋回事的BBS站友眉飞色舞的讨论C#里面可以不用指针等等则让我觉得好笑。C#就象当年的ASP一样，“忽如一夜春风来，千树万树梨花开”，结果许多学校的信息学院成了“Web 学院”。96,97级的不少大学生都去做Web了。当然我没有任何歧视某一行业的意识。我只是觉得如果他们把追赶这些时髦技术的时间多花一点在基础的课程上应该是可以走得更远的。几个误区 初学者对C#风潮的追赶其实也只是学习过程中经常遇到的几个误区之一。我将用一些实际的例子来说明这些现象，你可以按部就班的看看自己是不是属于其中的一种或者几种：认为计算机技术等于编程技术：有些人即使没有这个想法，在潜意识中也有这样的冲动。让我奇怪的是，许多信息学院的学生也有这样的念头。认为计算机专业就是编程专业，与编程无关的，或者不太相关的课程他统统都不管，极端的学生只要书上没带“编程”两个字他就不看。其实编程只是计算机技术应用过程中一种复杂性最低的劳动，这就是为什么IT业最底层的人是程序员（CODER）。计算机技术包括了多媒体，计算机网络，人工智能，模式识别，管理信息系统等等这些方面。编程工作只是在这些具体技术在理论研究或者工程实践的过程中

表达算法的过程。编程的人不一定对计算机技术的了解就一定很高。而一个有趣的现象是，不少大师级的计算机技术研究者是懂编程的。网上的炒作和现实中良好的工作待遇把编程这种劳动神秘化了。其实每一个程序员心里都明白，自己这些东西，学的时候并不比其它专业难，所以自然也不会高档到哪里去。 咬文嚼字的孔已己作风: 我见过一本女生的《计算机网络原理》教材，这个女生象小学生一样在书上划满了横杠杠，笔记做得满满的，打印出来一定比教材还厚。我不明白的是，象计算机网络原理这样的课程有必要做笔记？我们的应试教育的确害了不少学生，在上《原理》这一类课程的时候许多学生象学《马列原理》一样逐字背诵记忆。这乃是我见过的最愚蠢的行为。所谓《原理》，即是需要掌握它为什么这样做，学习why，而不是how（怎样做）。极端认真的学生背下以太网的网线最大长度，数据帧的长度，每个字段的意义，IP报头的格式等等，但是忘了路由的原则，忘了TCP/IP协议设计的宗旨。总之许多人花了大量的时间把书背得滚瓜烂熟却等于什么也没学。在学习编程的时候这些学生也是这样，他们确切的记得C语法的各个细节。看完了C教程后看《Thinking in C》（确实是好书），《Inside C》，《C reference》，this C, that C……，然后是网上各种各样的关于C语法的奇闻逸事，然后发现自己又忘了C的一些语法，最后回头继续恶补…。有个师弟就跟我说：“C太难了，学了这里忘了那里，学了继承忘了模板。”我的回答道：“你不去学就容易了”。我并没有教坏他，只是告诉他，死抠C的语法就和孔已己炫耀茴香豆的茴字有几种写法一样毫无意义。你根本不需要对的C语法太关心，动手编程就是了

，有不记得的地方一查MSDN就立马搞定。我有个结论就是，实际的开发过程中对程序语法的了解是最微不足道的知识。这是为什么我在为同学用Basic（我以前从没有学过它）写一个小程序的时候，只花了半个小时看了看语法，然后再用半个小时完成了程序，而一个小时后我又完全忘记了Basic的所有关键字。不顾基础，盲目追赶时髦技术：终于点到题目上来了。大多数的人都希望自己的东西能够马上跑起来，变成钱。这种想法对一个已经进入职业领域的程序员或者项目经理来说是合理的，而且IT技术进步是如此的快，不跟进就是失业。但是对于初学者来说（尤其是时间充裕的大中专在校生），这种想法是另人费解的。一个并未进入到行业竞争中来的初学者最大的资本便是他有足够的时间沉下心来学习基础性的东西，学习why而不是how。时髦的技术往往容易掌握，而且越来越容易掌握，这是商业利益的驱使，为了最大化的降低软件开发的成本。但在IT领域内的现实就是这样，越容易掌握的东西，学习的人越多，而且淘汰得越快。每一次新的技术出来，都有许多初学者跟进，这些初学者由于缺乏必要的基础而使得自己在跟进的过程中花费大量的时间，而等他学会了，这种技术也快淘汰了。基础的课程，比方数据结构，*作系统原理等等虽然不能让你立马就实现一个linux（这是许多人嘲笑理论课程无用的原因），但它们能够显著的减少你在学习新技术时学习曲线的坡度。而且对于许多关键的技术（比方Win32 SDK 程序的设计，DDK的编程）来说甚至是不可或缺的。一个活生生的例子是我和我的一个同学，在大一时我还找不到开机按钮，他已经会写些简单的汇编程序了。我把大二的所有时间花在了汇编，计算机体系结构

，数据结构，*作系统原理等等这些课程的学习上，而他则开始学习HTML和VB，并追赶ASP的潮流。大三的时候我开始学习Windows *作系统原理，学习SDK编程，时间是漫长的，这时我才能够用VC开发出象模象样的应用程序。我曾一度因为同学的程序已经能够运行而自己还在学习如何创建对话框而懊恼不已，但临到毕业才发现自己的选择是何等的正确。和我谈判的公司开出的薪水是他的两倍还多。下面有一个不很恰当的比方：假设学习VB编程需要4个月，学习基础课程和VC的程序设计需要1年。那么如果你先学VB，再来学习后者，时间不会减少，还是1年，而反过来，如果先学习后者，再来学VB，也许你只需要1个星期就能学得非常熟练。几个重要的基础课程 如果你是学生，或者如果你有充足的时间。我建议你仔细的掌握下面的知识。我的建议是针对那些希望在IT技术上有所成就的初学者。同时我还列出了一些书目，这些书应该都还可以在书店买到。说实在的，我在读其他人的文章时最大的心愿就是希望作者列出一个书单。大学英语 - 不要觉得好笑。我极力推荐这门课程是因为没有专业文档的阅读能力是不可想象的。中文的翻译往往在猴年马月才会出来，而现在的许多出版社干脆就直接把E文印刷上去。学习的方法是强迫自己看原版的教材，开始会看不懂，用多了自然熟练。吃得苦下得狠心绝对是任何行业都需要的品质。计算机体系结构和汇编语言 - 关于体系结构的书遍地都是，而且也大同小异，倒是汇编有一本非常好的书。《80x86汇编语言程序设计教程》（清华大学出版社，黑色封面，杨季文著）。你需要着重学习386后保护模式的程序设计。否则你在学习现代*作系统底层的一些东西的时候会觉得是在看天书。计

计算机*作系统原理 - 我们的开发总是在特定的*作系统上进行，如果不是，只有一种可能：你在自己实现一个*作系统。无论如何，*作系统原理是必读的。这就象我们为一个芯片制作外围设备时，芯片基本的工作时序是必需了解的。这一类书也很多，我没有发现哪一本书非常出众。只是觉得在看完了这些书后如果有空就应该看看《Inside Windows 2000》(微软出版社，我看的是E文版的，中文的书名想必是Windows 2000技术内幕之类吧)。关于学习它的必要性，ZDNET上的另一篇文章已经有过论述。

数据结构和算法 - 这门课程能够决定一个人程序设计水平的高低，是一门核心课程。我首选的是清华版的(朱战立，刘天时)。很多人喜欢买C版的，但我觉得没有必要。C的语法让算法实现过程变得复杂多了，而且许多老师喜欢用模块这一东西让算法变得更复杂。倒是在学完了C版的书以后再来浏览一下C的版的书是最好的。

软件工程 - 这门课程是越到后来就越发现它的重要，虽然刚开始看时就象看马哲一样不知所云。我的建议是看《实用软件工程》(黄色，清华)。不要花太多的时间去记条条框框，看不懂就跳过去。在每次自己完成了一个软件设计任务(不管是练习还是工作)以后再来回顾回顾，每次都会有收获。

Windows 程序设计 - 《北京大学出版社，Petzold著》我建议任何企图设计Windows程序的人在学习VC以前仔细的学完它。而且前面的那本《Inside Windows 2000》也最好放到这本书的后面读。在这本书中，没有C，没有GUI，没有控件。有的就是如何用原始的C语言来完成Windows程序设计。在学完了它以后，你才会发现VC其实是很容易学的。千万不要在没有看完这本书以前提前学习VC，你最好碰都不要碰。我知

道的许多名校甚至都已经用它作为教材进行授课。可见其重要。上面的几门课程我认为是必学的重要课程（如果你想做Windows程序员）。对于其它的课程有这样简单的选择方法：如果你是计算机系的，请学好你所有的专业基础课。如果不是，请参照计算机系的课程表。如果你发现自己看一本书时无法看下去了，请翻到书的最后，看看它的参考文献，找到它们并学习它们，再回头看这本书。如果一本书的书名中带有“原理”两个字，你一定不要去记忆它其中的细节，你应该以一天至少50页的速度掌握其要领。尽可能多的在计算机上实践一种理论或者算法。你还可以在CSDN上阅读到许多书评。这些书评能够帮助你决定读什么样的书。日三省乎己 每天读的书太多，容易让人迷失方向。一定要在每天晚上想想自己学了些什么，还有些什么相关的东西需要掌握，自己对什么最感兴趣，在一本书上花的时间太长还是不够等等。同时也应该多想想未来最有可能出现的应用，这样能够让你不是追赶技术潮流而是引领技术潮流。同时，努力使用现在已经掌握的技术和理论去制作具有一定新意的东西。坚持这样做能够让你真正成为一个软件“研发者”而不仅仅是一个CODER。把最多的时间花在学习上 这是对初学者最后的忠告。把每个星期玩SC或者CS的时间压缩到最少，不玩它们是最好的。同时，如果你的ASP技术已经能够来钱，甚至有公司请你兼职的话，这就证明你的天份能够保证你在努力的学习之后取得更好的收益，你应该去做更复杂的东西。眼光放长远一些，这无论是对谁都是适用的。相信你已经能够决定是否学习C# 或者什么时候去学它了。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