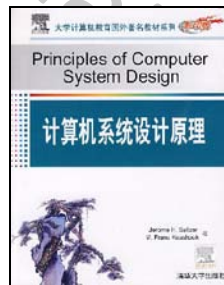
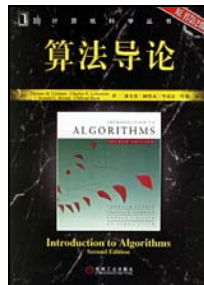
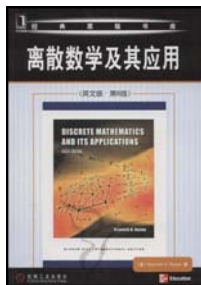


附录 B 国内外著名高校计算机学科课程体系

自 20 世纪 40 年代世界第一台现代计算机在美国诞生以来，美国一直执全球计算机学界之牛耳，这同时也是美国计算机产业界占据绝对优势的重要原因之一。我们成批量地引进的国外众多优秀教材绝大多数也都来自美国。计算机学科仍然在高速发展，与此对应的计算机人才培养模式也在不断变化，密切关注和跟踪国外尤其是美国名校的教学新动态乃至培养思路和教育思想，应该是非常有意义的。本附录选择了美国计算机学科最负盛名的五所高校以及国内的两所著名高校，对目前各校计算机科学专业的本科教学体系进行介绍，供广大读者参考。

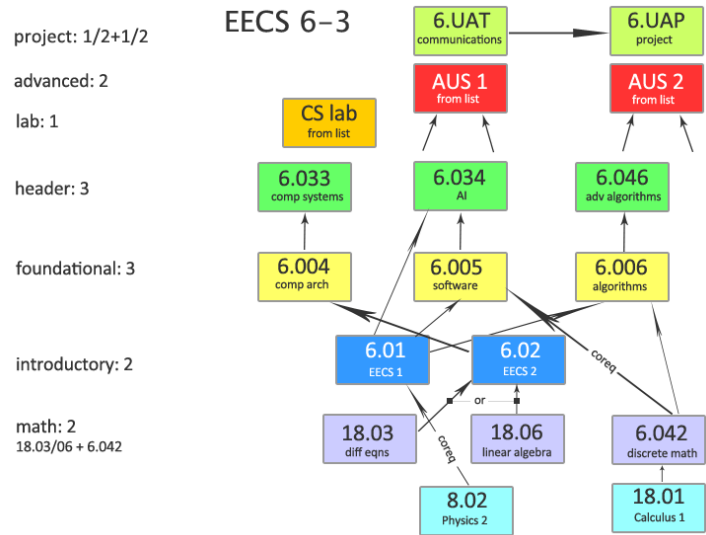
1. Massachusetts Institute of Technology (MIT, 麻省理工学院)

MIT 没有典型意义上的计算机专业，偏软的只有理论计算机科学和人工智能及其应用两个专业。因此没有类似于其他学校的导论课程。在 MIT 的计算机科学系中，主要有以下几门主干课程：6.001 “计算机程序的构造和解释”，采用的是 Harold Abelson 等编著的教材 *Structure and Interpretation of Computer Programs* (Second Edition)；6.046 “算法导论”；6.033 “Computer Systems Engineering”（采用的教材为《计算机系统设计原理》）；6.004 “计算结构”（Computation Structures）（自编课件）。此外有两门专业基础数学课：“概率系统分析”（教授自编教材）和“计算机科学数学”，后者的教材是国外院校普遍采用的 Rosen 所著《离散数学及其应用》。



对 MIT 的学生而言，实验课程有多种选择：电气工程和计算机科学实验，模拟电子

实验，数字系统实验，微机项目实验，半导体设备项目实验。此外，无论何种专业，都有软件工程实验课。值得注意的是，本科生各专业的必修课程中并没有软件工程课程。也就是说，软件工程的内容都在实践中完成了。MIT 的本科生课程体系如下图所示。



2. Stanford University (斯坦福大学)

Stanford 大学拥有独立的计算机科学系。浏览该校的教学手册，最具特色的恐怕要算多门科普性计算机知识讲座了，一般有两到三个单元，涉及面非常之广，从量子计算到数字演员，从计算科学的伟大思想到网络安全，从电子商务到大型开源软件的分析，其中还不乏对计算思维、计算机面临的困境以及迅速发展所带来的诸多问题的思考。开课的老师阵容强大，基本上都是响当当的名教授，甚至包括图灵奖得主，例如 John McCarthy。用这种讲座代替计算机科学导论性质的专门课程，可以充分展示计算机科学的丰富内涵，使学生较早期地了解学科的轮廓和脉络，对于开阔学生视野，启发学生的学习兴趣也大有好处。该校的著名教授 Eric Roberts 曾执教入门类课程多年，总结了一套在语言教学中融入软件工程和现代程序设计观念，结合算法和数据结构教学的经验。其成果就是《C 语言的科学和艺术》和《C 程序设计的抽象思维》两本书（影印版已由机械工业出版社出版）。

作为 CC2001 工作组两位主席之一，他在 C 语言教学中强调库与接口设计、编程风格的重要性，并进而介绍抽象、封装的概念，产生了很大影响。

Stanford 大学计算机本科核心课程如下：

(1) System 方面

- cs106B: Programming Abstractions
- cs107: Computer Organization and Systems (教材为《深入理解计算机系统》)
- cs110: Principles of Computer Systems (教材同上)

(2) Theory 方面

- cs103: Mathematical Foundations of Computing
- cs109: Introduction to Probability for Computer Scientists
- cs161: Design and Analysis of Algorithms (教材为 MIT 的《算法设计》)



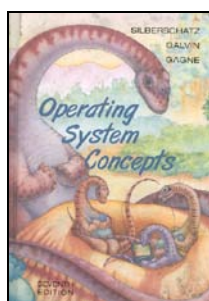
3. Carnegie Mellon University (CMU, 卡内基梅隆大学)

与 MIT、Berkley 等学校计算机科学和电子与电气工程同处一系不同，CMU 的计算机科学系成立于 1965 年，是全美最早的，并随后升格为计算机科学学院。其研究生项目中除了机器人方向与硬件关系较多之外，其他基本上都是纯软的。从这个意义上来说，CMU 的教学体系对于偏软的计算机科学系有较大的借鉴意义，例如国内各高校成立的众多软件学院。

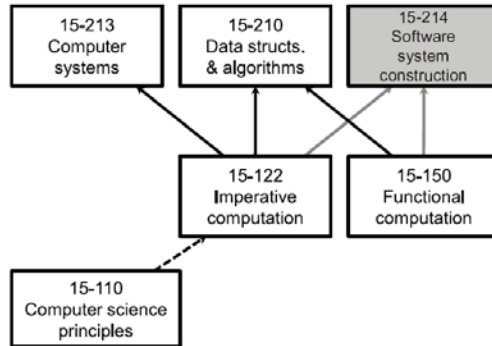
CMU 的教学手册上没有从传统意义上针对计算机专业学生的导论课，其中名为

Great Theoretical Ideas in Computer Science 的两学期课程作为离散数学的替代。此课程没有教材，内容比传统离散数学要灵活得多，涉及概率、代数、算法、加密理论、复杂性理论、博弈论等，非常注重学习的趣味性和实用性。与其他名校相同，CMU 对程序设计的重视也给人留下很深印象：新生的第一堂课就是“初中级程序设计”，直接讲授 Java。然后是中高级程序设计（Java）、C 语言编程技巧、高级编程实践（Java）、程序设计原理（用 SML 语言讲授）。

目前计算机专业教学计划中的一个难点，是硬件课程的设置问题。硬件知识体系本身非常丰富，但是硬件课程多了，又削弱了计算机专业的特色。CMU 在这一问题上是怎样处理的呢？计算机学院的现任院长 Randal E. Bryant 亲自给出了回答，他用“计算机系统导论”一门课（12 个单元）完成了硬件知识的教学。这项教学改革的结果就是一本厚达 900 多页的书：*Computer Systems: A Programmer's Perspective*（中译本《深入理解计算机系统》）一书。他在该书的序言中说：“本课程的宗旨是用一种不同的方式向学生介绍计算机。因为，我们的学生中几乎没有人有机会构造计算机系统。而大多数学生，甚至是计算机工程师，也要求能日常使用计算机和编写计算机程序。所以我们决定从程序员的角度来讲解系统，并只讨论那些影响用户级 C 程序的性能、正确性或实用性的主题。这种做法使得我们讲授本课程的方式对学生来讲既实用、具体，又能实践，同时也非常利于调动学生的积极性。”网站上的一些随书配套实验也独具匠心。因此此书的成功是水到渠成的，它已经被全球近百所院校采用作为教材。



CMU 大学计算机本科核心课程如下：



(1) Computer Science Core

- 15-122: Principles of Imperative Computation
- 15-150: Principles of Functional Programming
- 15-210: Parallel and Sequential Data Structures and Algorithms
- 15-213: Introduction to Computer Systems
- 15-251: Great Theoretical Ideas in Computer Science
- 15-451: Algorithm Design and Analysis

(2) One Algorithms & Complexity elective

- 15-354: Computational Discrete Mathematics
- 15-355: Modern Computer Algebra
- 15-453: Formal Languages and Automata
- 21-301: Combinatorics
- 21-484: Graph Theory

(3) One Application elective

- 05-391: Designing Human-Centered Software
- 05-431: Software Structures for User Interfaces
- 10-601: Machine Learning
- 11-411: Natural Language Processing

- 15-313: Foundations of Software Engineering
- 15-322/15-323: Introduction to Computer Music
- 15-381: Artificial Intelligence: Representation and Problem Solving
- 15-384: Robotic Manipulation
- 15-385: Computer Vision
- 15-415: Database Applications
- 15-462: Computer Graphics

(4) One Logics and Languages elective

- 15-312: Foundations of Programming Languages
- 15-317: Constructive Logic
- 15-414: Bug Catching: Automated Program Verification and Testing
- 21-300: Basic Logic
- 80-311: Computability and Incompleteness

(5) One Software Systems elective

- 15-410: Operating System Design and Implementation
- 15-411: Compiler Design
- 15-418: Parallel Computer Architecture and Programming
- 15-440: Distributed Systems
- 15-441: Computer Network

4. University of California-Berkeley (加州大学伯克利分校)

Berkeley 的课程设置也有很多独树一帜的地方，尤其是在专业基础课方面，除了有专业导引课程“计算机科学专题”之外，对于没有编程经验的学生，第一门编程课是符号编程入门，采用 LISP 语言。有一定编程经验或者有自学能力的学生，可以选择多种语言 and 环境的自主学习课程，包括 C、Fortran、C++、Java，以及 UNIX 的使用等，这种多元化与 Berkeley 计算机科学与电子电气工程同系有关。但是所有学生在第二学期都要学习一

组基础课：61A“计算机程序的构造和解释”，采用教材同 MIT 的 6.001 课程一样；61B“数据结构”（教材采用自编讲义）；61C“计算机结构”（Machine Structures），采用 Hennessy 的《计算机组织与设计》（中译本清华大学出版社出版，机械工业出版社出版了影印版）。这些课程蕴涵了 Berkeley 多年的教学经验结晶。



Berkeley 其他比较有特点的课程还有：将离散数学和概率论结合讲授的 CS70，主讲是名教授 Christos Papadimitriou；CS98-1 编程练习课，以大学生编程竞赛中的赛题为授课素材；CS 169 软件工程直接用 Kent Beck 的《极限编程》（人民邮电出版社出版了中译本）作为教材。除了软件工程课程常见内容外，教学侧重实际，贯穿了极限编程的思想，涵盖 UML、JUnit 单元测试、软件架构、设计模式和反模式、重构、CVS 版本控制、系统和集成测试，最后要求完成一个实际产品，并进行演示。

Berkeley 大学计算机本科核心课程如下：

(1) Core

- CS 61A: Structure and Interpretation of Computer Programs
- CS61B: Data Structures
- CS61C: Machine Structures
- EECS42: Digital Electronics

(2) Upper Division Required

- CS 170: Algorithms
- CS 162: Operating systems

(3) Breadth courses choose two from the following

- CS 150: Digital Systems
- CS 152: Computer Architecture
- CS 160: User Interfaces
- CS 161: Computer Security
- CS 164: Languages and Compilers
- CS 169: Software Engineering
- CS 184: Computer Graphics
- CS 186: Databases
- CS 188: Artificial Intelligence

5. University of Illinois-Urbana-Champaign (伊利诺依大学香槟分校)

UIUC 的计算机专业创建于 1972 年，到 1986 年基本定型，其教学体系中的课程也是相当丰富。其中，本科核心课程如下：

(1) 数值分析方向

- Math225: 矩阵论
- CS257: 数值方法
- CS35x: 包括数值分析、常微分数值方法、数值线性代数

(2) 理论方向课程

- CS173: 离散结构
- CS273: 计算理论
- CS37x: 包括算法、形式方法、程序验证

(3) 人工智能方向课程

- CS348: 人工智能导论
- CS34x: 包括机器人、机器学习和模式识别

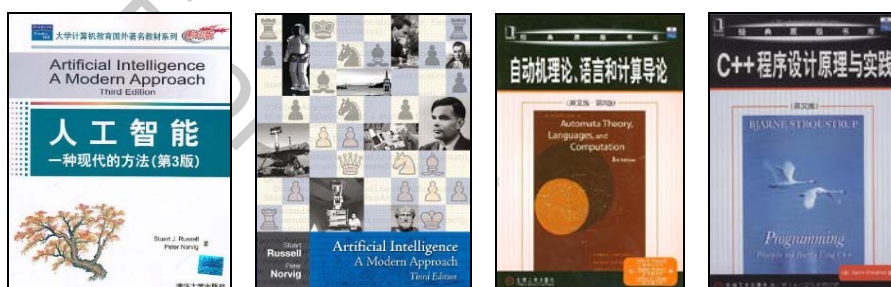
(4) 软件方向

- CS125: 计算机科学导论

- CS225: 数据结构与软件工程原理
- CS31x: 包括数据库、图形学、多媒体
- CS32x: 包括软件工程、操作系统设计、分布式系统、编程语言与编译器、并行计算、实时系统、编译器构造、编程语言设计

(5) 硬件方向课程

- CS231/ CS232: 计算机体系结构
- CS33x 包括计算机组成、VLSI 系统与逻辑设计、通信网络、嵌入式体软件



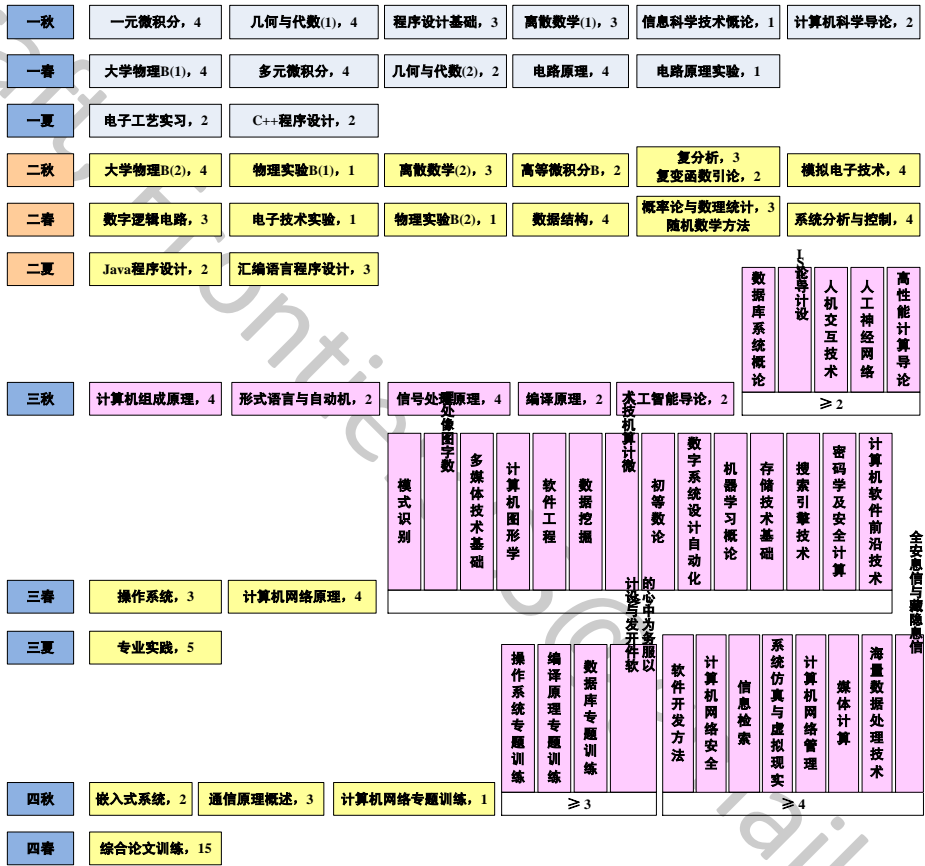
与 Stanford 大学不同的是，UIUC 的计算机科学导论课程比较简单，只有一门为新生开的计算机科学导向课（CS100），而且并非必修。名为“计算机科学导论”的 CS125 实际上是以 Java 语言为主的编程入门课，涵盖了一些算法的内容。此外还有与之配套的实验课。当然，系里所开的许多面向高年级和研究生层次的讲座是对低年级开放的。

6. 清华大学—计算机科学与技术系

清华大学早在 1956 年就建立了电子计算机专业，1957 年即培养出我国第一批计算机专业毕业生。经过半个世纪的不懈努力，已发展成为我国教学、科研综合实力优势突出的计算机系，2006 年在全国计算机科学与技术学科以及学科整体水平评估中排名第一。

2003 年清华大学计算机系对本科生培养方案和课程大纲作了较大调整，更加突出“宽口径、厚基础”的培养思想，同时汲取了“IEEE/ACM 计算机课程体系 2001”（IEEE/ACM Curricula 2001）的有益养分。本科专业按照学分制管理机制，实行弹性学习年限，学生在获得培养方案所要求的不少于 172 学分后即可毕业。本科学制分为四年，部分学生可

提前完成学业。教学采用春、秋、夏三学期制，春、秋学期各 18 周，为正常课程教学学期，夏季学期一般 5 周，用于实践训练教学。清华大学计算机科学与技术系每学年的具体课程如下图所示。



7. 上海交通大学—软件学院

上海交通大学软件学院于 2001 年 5 月正式成立。学院借鉴 ACM/IEEE 的软件工程本科教育体系 (SEEK)，参考 IEEE 的软件工程知识体系 (SWEBOK)，结合国内软件产业的紧迫需求，逐步形成了较为科学、完善、具有软件工程专业特色的本科生教学体系。软件学院本科培养计划分三阶段实施：一、二年级：夯实数理基础、完善程序设计能力；三

年级：掌握学科基础、培养系统设计开发能力；四年级：顶峰体验、项目实践，拓展领域知识。上海交通大学软件学院每学期的部分核心课程如下表所示。

学年	上学期	下学期
一	高等数学（上） 线性代数 程序设计	高等数学（下） 普通物理（上） 计算机系统基础（上）
二	普通物理（下） 计算机系统基础(下) 数据结构与算法 Great Theoretical Ideas	计算机系统工程 算法设计与分析 并行计算 数据库设计
短学期	软件课程	
三	概率论（必修） 数字部件设计 操作系统	编译原理 计算机网络 计算机体系结构
四	项目实践	毕业设计

8. 分析与总结

国外著名高校的计算机学科中全专业必修的专业基础课特点包括：数量少、基础宽、覆盖面广，综合性强（面向问题），前沿性，对教师要求高，实践量大（Learning by doing），大量的实验（lab 和 project），阅读量大，作业和实验需要自学很多内容，有些课程本身也要求大量阅读。归纳起来，有这么几个特点，值得我们学习和思考：

- 硬件课程整体在减少

偏软类的三所院校中，CMU 最为彻底，硬件课程只有一门课，而 UIUC 也只有两门必修（两门体系结构），Stanford 也是两门（电子学、体系结构或数字系统）。其中的原因，前面引述的 Randal E. Bryant 所言作出了解释，毕竟计算机科学需要关注的在计算机系统层次中已经越来越高，底层越来越变得透明了。事实上，CC2001 中制定的硬件课程也只有一门。而国内目前一般还开设数字逻辑、微机原理、计算机组成与结构、微机实验等

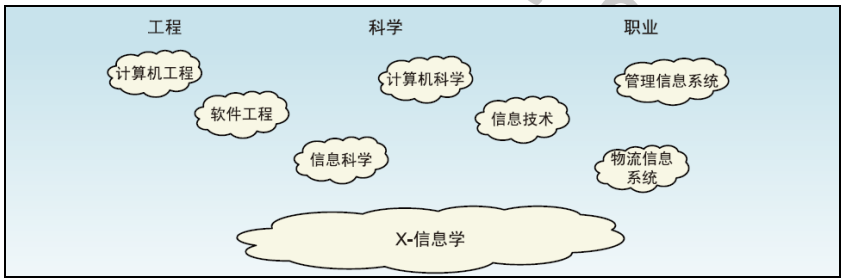
硬件类课程。

- 程序设计日益重视

在 CMU、UIUC 和 Stanford 大学，必修的程序设计类课程往往在四五门左右。Berkley 加起来也有四门。MIT 虽然没有大量前导性的编程课程，但是由于在后面计算机系统工程、计算机语言工程、软件工程实验、Web 软件工程诸课程都有实际的项目要完成，所以实际学时也很多，UIUC 的改革更说明了这一点。同时，还出现了强调提高程序设计技巧，与软件工程环境和工具相结合，提倡团队合作，高级程序设计课程与数据结构、算法课融合的趋势。

- 计算机跨学科交叉课程的重视

国外的众多名校他都十分注意计算机学科与其他相关学科的交叉与融合，并在不断扩展计算机学科领域。新的计算机科学应该是在数学与社会学支持下，与应用领域（如其他科学、人类社会、艺术、商务服务等）相结合的新型学科。新型计算机学科应包括计算机科学核心基础、所需要的相关科学（如物理、离散数学、认知科学、社会学、经济学和生物学等）以及专业构成（如计算机工程、软件工程、信息技术应用）等部分。此外，它还应具有广泛的信息学应用领域。新型计算机学科将带来全新的“计算思维”。



(X = art, astro, bio, business, chem, eco, game, geo, health, media, medical, social...)

应该说，这些名校的教学体系、教材和经验都是丰富的宝藏，值得好好挖掘，学习每门核心课程的教学法，实验课程的安排，各门课程的衔接，大型项目的选材等等方面。