美国高校第一门计算机课程调研情况分析

李 波

摘 要：通过参加 Microsoft Research Faculty Summit 2013 会议，并实地考察了西雅图大学、斯坦福大学和加州大学伯克利分校，结合网上课程调研，着重分析了美国高校第一门计算机课程的基本情况，并与我国高校非计算机专业第一门计算机课程进行了对比，讨论了第一门课程设计中涉及的关键问题和相关技术争论。试图为我国高校非计算机专业的第一门计算机课程的设立和教学改革实践提供国际经验。

关键词：美国高校；第一门计算机课程；CS1；调研；对比

一、第一门计算机课程的界定

美国高校一般将第一门计算机课程称之为入门课系列，课程编号通常为 CS1 或 CS101，部分高校以计算原理为主要授课内容，部分高校以程序设计为主要授课内容。国内高校非计算机专业的主流课程体系为 1+X 方案，第一门计算机课程为大学计算机基础，X 包括程序设计等多门课程。故美国的 CS1 可相应地对应国内的大学计算机基础和程序设计两门课程。

二、调研途径

本次调研的途径有三：(1) 教指委相关成员应邀参加 Microsoft Research Faculty Summit 2013 会议；(2) 实地访问西雅图大学、斯坦福大学和加州大学伯克利分校；(3) 网上课程及 MOOC 调研。

Microsoft Research Faculty Summit 2013 会议于 2013 年 7 月 15、16 日在微软总部 Redmond 举行，该会议围绕深度学习（Deep learning）、大数据（Big Data）和云计算（Cloud computing）三大主题在研究和教学方面进行研讨。其中有两个与教学相关的专题会议：计算机辅助教学（Computer-Aided Education），面向和通过浏览器的现代编程（Modern Programming for or via Web browser）。另外附设了一个展示会也有微软研究院在计算机辅助教学方面的研究成果。计算机辅助教学的研究在大规模开放在线课程的背景下需要更多的研究和探索，该领域是一个研究热点。面向和通过浏览器的现代编程专题会议分别就面向 Web 的编程和通过 Web 浏览器进行编程报告了相关研究成果，在发明新语言（Type Script）来简化 Web 编程和面向移动设备（Touch Develop）支持学生随时随地编程取得了进步。在附设的展示会微软研究院介绍了.NET Gadgeteer 和 Lab of Things 两项产品，该产品支持学生进行硬件及物联网方面的设计和开发。在会议期间与 Coursera 的联合创始人斯坦福大学的 Andrew NG 教授进行了会谈，了解了 MOOC 的现状和 Coursera 的运作模式。在会议结束之后分别实地考察了西雅图大学、斯坦福大学和加州大学伯克利分校。了解了其课程开设情况和课程评估情况。

网上课程及文献研究，选择了 10 门知名美国大学典型的入门课程进行了系统化的研究，这 10 门课程如下：

MIT 6.00: Introduction to Computer Science and Programming；

Stanford CS106A: Programming Methodology；

Stanford CS106B: Programming Abstractions；

李 波，西安交通大学计算机教学实验中心副主任，副教授。

Stanford CS107: Programming Paradigms；

CMU CS-15110: Principles of Computing；

UC Berkeley EECS | CS10: The Beauty and Joy of Computing；

Princeton Computer Science 116: The Computational Universe；

Princeton Computer Science 109: Computers in Our World；

Harvard CS50: Introduction to Computer Science；

Harvard CS101: From Nand to Tetris.

为了评价 MOOC 对计算机基础课程的影响, 另外在课程调研中增加了 Udacity 的 CS101 的调研, 了解 MOOC 在第一门计算机课程的应用情况。

三、CS1 开设情况

通过对样本的分析对比，主要选择最有代表性的卡内基梅隆大学（CMU CS-15110）、加州大学伯克利分校（UC Berkeley CS10）、斯坦福大学（Standford CS101）和大规模在线课程 Udactiy（Udacity CS101）的四门入门课程进行调研。

卡内基梅隆大学的计算机科学学院在 2011 年发布对其入门课程系列的改革方案, 并从 2012 年开始实施。其入门课程系列包括三门课程：CS-15110（Principles of Computing），CS-15122（Principles of Imperative Computation）和 CS-15150（Principles of Functional Computation）。主要的改变有三个方面的考虑：(1）对计算机专业和非计算机专业学生在入门课程中如何体现计算思维；(2）强调软件系统的高可靠性（highly reliable）及其实现方式；(3）强调并行计算及编程的内容。这里重点讨论计算思维方面的情况：计算思维既对数学、科学、工程等学科有支撑作用, 又对人文、艺术和商业等学科产生巨大影响。不但计算机专业学生要掌握，非计算机专业的学生更要应知应会。对于非计算机专业的学生入门系列课程要起到双重作用：掌握实用的计算机科学的应用技能和计算思维的坚实基础，期望在其今后的职业生涯中能获得新的技能。

CMU 的 CS-15110 课程名称为“计算原理”（Principles of Computing），每届有 900 学生左右，面向所有专业，包括计算机专业。共 15 周，每周 3 节课（1 小时）和 1 节实验（2 小时），课外安排 5 小时。考核办法为：平时作业占 30%、实验参与占 5%、两次实验考试占 10%、平时书面考试占 30%、期终书面考试占 25%。从其考核安排可以看出其考试频繁，作业和实验工作量大，特别重视课程的过程考核。该课程是面向很少或没有计算基础知识的学生，以阐述基本计算原理为重点的课程，涵盖以下主题：编程构造（排序、选择、迭代、递归），数据组织（数组和列表），抽象（数据表示、计算机组织、计算机网络、功能分解和应用程序编程接口），问题求解的计算原理（分而治之、随机、并发），计算理论（复杂性、不可计算函数、启发式解决复杂问题、计算问题的分类）以及与计算机科学相关的社会、伦理和法律问题。教材选用俄勒冈州立大学 John Conery 教授编写的 *Explorations in Computing: An Introduction to Computer Science*（CRC Press, 2011, ISBN: 978-1439812624）。参考书为 MIT 的 Hal Abelson 教授等编写的 *Blown To Bits: Your Life, Liberty, and Happiness after the Digital Explosion*（Addison-Wesley, 2008, ISBN: 978-0137135592）。该课程的具体内容见表 1，实验安排见表 2。

UC Berkley 的入门课程是 CS10，课程名称是“计算之美、计算之乐”(The Beauty and Joy of Computing)。此课程专门面向为非计算机专业，主要的内容包括三个部分：(1）如何将解决问题的想法变成计算机程序（Big Ideas of Programming），主要包括：抽象、算法、递归、编程范型、并发、分布式计算 ;(2)计算机学科的重要思想(Big Ideas of Computing)，主要包括：3D 图形、视频游戏、计算博弈论的原理、人工智能、人机交互、云计算、计算的极限、计算对世界及社会的影响；(3）通过丰富的计算机应用阐明计算机如何改变世界（Beauty and Joy），这部分的内容通过完成 CS Unplugged 项目的活动和两个同学一组的进行一个三个周的结对编程项目来完成此部分的教学。本课程采用 Scratch 语言进行教学, Scratch 是易用

的图形化的编程语言，具有模块化的程序构造和采用在界面上进行拖拽方式进行编程的特点。为了使学生得到软件工程的训练，在编程开发过程中采用结对编程（Pair Programming）的模式进行。

表 1　CMU CS-15110 教学内容

周次	每周第一讲	每周第二讲	每周第三讲
Unit 01: History of Computing	Overview	Pre-Electronic Computing	Electronic Computing
Unit 02: Introduction to Ruby	Basics	For Loops	-
Unit 03: Algorithmic Thinking	Ranges, Arrays and Iterators	Arrays, Sieve of Eratosthenes	-
Unit 04: Iteration	Linear Search	Insertion Sort	Introduction to Big O
Unit 05: Recursion	Recursive Thinking	Binary Search	Merge Sort
Unit 06: Organizing Data	List-Based Data Structures	Hash Tables	Non-Linear Data Structures
Unit 07: Data Representation	Integer, Floating Point, Text	Compression	Images and Sound
Unit 08: Computer Organization	Boolean Logic, Gates	Levels of Abstraction	The Machine's Language
Unit 09: Randomness	Random Number Generators	Games with Random Numbers	More Fractals and Cellular Automata*
Unit 10: Concurrency	Multitasking/Deadlock		Pipelining/Distributed Computing
Unit 11: The Internet	Design Principles	Layers and Abstraction	Encryption
Unit 12: Simulation	Basic Concepts, Example	Continuous-Time Simulations	-
Unit 13: Artificial Intelligence	Games and Search Strategies	Natural Language Processing	Smartest Machine on Earth(NOVA video) Watson wins Jeopardy!
Unit 14: The Limits of Computing	Intractability	P and NP	Non-computability
Epilogue: The Future of Computing	Quantum Computing		

表 2　CMU CS-15110 实验内容

Lab 1	Lightbot
Lab 2	Intro to Ruby and irb
Lab 3	Arrays and Loops
Lab 4	More Searching and Sorting
Lab 5	Debugging Practice
Lab 6	Fractals
LAB EXAM 1	
Lab 7	Hash Tables
Lab 8	Bitmap Images
Lab 9	Cellular Automata
Lab 10	Review
Lab 11	Graphics in Ruby
Lab 12	Programming a Game
LAB EXAM 2	

Stanford 大学的 CS101 课程名称为“ 计算基础 ”（Essentials of computing），课程的主要教学目的是让学生了解和理解计算机的工作原理、优点及缺点。主要内容包括编程、数字化、安全、网络。教学方式主要采用在线课程网站辅助的自学方式。编程语言采用 JavaScript，无特殊编程工具，所以编程实验全部在浏览器中进行。教学内容包括：计算机及代码的特性、哪些问题计算机可以解、哪些问题计算机不能解；计算机硬件（芯片、CPU、内存、硬盘）的工作原理；计算机软件的工作原理，什么是程序、什么是运行；数字图像、视频工作原理；编程基本概念；如何组织数据；因特网工作原理（IP 地址、路由、以太网、WIFI）、计算机安全。

MOOC 平台 Udacity 的 CS101 课程名称为“ 计算机科学概论 ”。本课程以开发一个搜索引擎为目标介绍计算机科学的基本概念及如何掌握程序设计，编程语言采用 Python 语言。设计搜索引擎的主要设计的问题是：抽取网页中的链接（如何从一个网页找出所有链接），搜索及收集网页（如何管理数据），建立逆向索引完成查询（使用复杂数据结构-图、哈希表），对搜索结果进行排序（如何使用递归）。教学内容以开发搜索引擎原型为主线介绍了程序设计的基本概念和 Python 的模块、类型、类、继承、文件及异常。

通过上面四门课程和实地的考察可知美国高校在第一门计算机课程的开设最显著的特征是多样性。多样性体现在有多种课程类型，主要课程类型分为以程序设计为核心（Programming Focus）和不以程序设计为核心（Programming de-focused）的两大类。程序设计类根据编程语言及编程范型又有命令式优先（Imperative-first）、面向对象式优先（Objects-first）和函数式（Functional-first）三种。非程序设计类分为宽度优先（Breadth-first）、算法优先（Algorithms-first）和硬件优先（Hardware-first）三种。多样性还体现教学内容选取的侧重上，有的课程注重知识性、有的课程注重原理性。

另外在第一门计算机课程的设计方面有的高校区分了专业与非专业，有的则没有区分；在课程设计上有的考虑到有后续课程，有的无考虑；课程的性质基本都是选修，只是考虑强调了课程的先修课程要求。

四、中美第一门计算机课程对比

为了清晰地分析清楚中美第一门计算机课程的异同，可以从 10 个方面进行对比形成如下表格。在课程内容设置方面美国第一门的课程从 Great principle of computing（伟大的计算原理）、Great Ideas in Computer Science（计算机科学的伟大思想）等角度出发强调课程的科学性及原理性，而中国的课程内容强调的是知识性和工具性。例如在讲授差分机这一知识点时，美国的课程通过实例讲解了差分机的求解方法，而中国大多数教科书只是介绍了差分机的发明过程，同学不会了解其工作原理。在课程要求方面，美国的课程提出了较高要求，例如在 CS1 中要求同学开发一个搜索引擎原型，并围绕此采用进行教学设计，学生觉得与自己熟悉的搜索有直接联系，并觉得很神秘、很复杂，通过学习自己能完成这样一个系统，很有成就感，教学也有“抓手”。在课外环节方面，CMU 的 CS-15110 及伯克利的 CS10 安排了非常细致和大量的作业和实验，配备了庞大的助教队伍，能有针对性地辅导学生及批改作业。在教学理念的物化方面美国的教师发明了许多支持第一门计算机课程的适合初学者的工具、语言，例如 Alice、Scratch、Raptor 等。具体情况见表 3。

表 3　中美第一门计算机课程对比

比较项目	美国大学	中国大学	结论
计算思维的影响	多样，普遍认识是像计算机科学家一样思考	共识、概念热	领先
课程设置思路	多路径（Pathways、Threads）实现多样性	分类分层次	相似
课程模式、风格	范型明确、六范型	有三种初步类型（何钦铭教授提出）	差距中
内容设置	原理性、科学性、层次分明 Great Ideas in Computer Science	知识性、工具性、层次不清（信息素养）	差距中
教学要求	要求高、很具体、有抓手，是重要课程	要求低、较抽象、难操作，学生称之为“水课”	差距大
课外环节	要求扎实并能及时给学生反馈	助教环节薄弱	差距大
考核	注重过程	偏重笔试	差距大
教材	特色明确、原创、数量少、他用多	特色不明确、抄编、数量多、自用多	差距大
教学效果评估	科学、第三方研究、引入第三方评价	不合理、自评为主	差距大
教学理念的物化	发明开发了很多工具、语言	没有，也不重视	差距大

五、CS1 设计的技术争论

在第一门计算机课程的设计中在很多技术方面长期存在不同的做法和各式各样的意见，并且长期存在着争论和争鸣。这反映了第一门计算机课程的多样性和计算机学科的复杂性，也反映了计算机学科发展的快速性。显然这些争论还将继续进行，我们要对这些问题有清晰的认识和自己独立的观点，进行合理的折中和取舍，从而指导

我们的课程设置及具体教学。争论主要体现在：基于传统平台还是新平台、编程范型与语言的选择、如何引入软件开发实践、如何对待并行处理、实验平台是在线（Via Web Browser）编译还是独立编译。这方面的大部分内容在 ACM 的 CC2001 及 CC2013 报告中有详细的介绍。

在CS1的教学中如何选择平台是一个困扰许多教师的问题。基于传统的平台选择 Windows 还是 Linux。随着编程装置（programmable devices）多样性迅速增长，例如 Web、Cloud、移动终端（手机）、robots、游戏终端、开源硬件（raspberry-pi、Arduino）都可作为教学平台。移动终端（手机）通过 jbit、C4droid、TouchStudio、Alogo 等工具也可成为主流的教学平台。所以在 CS1 中平台的选择存在多样性的可能，应该结合教学内容，合理地选择平台。

编程范型 Programming Paradigms（styles）主流的包括 Procedural、Modular、Data Abstraction、Object-oriented、Generic 等形式。编程语言可分为命令式、面向对象式、函数式、逻辑式，从编译和运行的角度语言又有静态、动态之分。另外编程范型和编程语言之间的关系十分复杂，而往往一个编程语言可以支持多种范型，但一个程序员很难驾驭多个编程范型，例如可以用 C++写出一个完全过程化的程序，可以用 C++写出一个纯粹的面向对象程序，甚至还有人可以写出糅杂了两种范型的程序。教师尤其是教材的编写者要深入地了解编程范型形成的动机和背景，才能更好地指导学生进行程序设计，编出的教材才能不至于出问题。例如在 C++中，作者一定要区分一些基本问题：为什么要用面向对象，什么时候它的价值才能体现，什么是基于对象和面向对象，ADT 不是面向对象。回顾近 20 年在 CS1 中教学语言选择方面呈现的“ 简单化 ”趋势：（1）语言更加安全，例如出现了 Java 和 C#等比 C 和 C++更加安全的 managed languages。（2）出现了大量的如 Python、Ruby、JavaScript（JQuery、Prototype.js）等动态脚本化的语言（dynamic languages，Script）作为教学语言。（3）可视化编程引入教学，例如 Alice、Scratch、Logo、流程图工具-Raptor 等都是可视化编程语言，有的就是专门为教学设计的语言。（4）高层抽象、无状态、描述性的语言如 Haskell、Scheme 、Snap 被引入教学。（5）基于库的支持，例如 University of Washington 的 Kelvin Sung 教授在 NSF 的支持下开展了 Game-Themed CS Education: Empowering the Faculty 项目，其目的是面向已有的 CS1 课程开发出进行游戏开发的基本库，既简化了学生开发的难度，又激发了学生的兴趣。该项目的成果在许多学校进行了应用，取得了较好的效果。（6）简化语法，例如 CMU 15122 课程采用的 C0 语言就是对 C 语言进行了语法简化，是一个“ Syntax light ”的语言。其实以上的趋势除了“ 简单化 ”之外，实质上是更好地体现了计算思维，周以真指出计算思维是“ Mental not Metal ”，其本质是要加强学生的思维能力训练而不是实现细节的培养。这样可以帮助学生跳过了语法训练细节，快速进入算法思考及设计。因此在 CS1 编程范型与语言的选择难点上要认真思考、充分认知、反复实践、合理选择。

软件工程及实践为我们提供了丰富的养分，但如何引入软件开发实践也是较为困难的选择。例如许多课程采纳了单元测试（测试驱动开发）、版本控制、结对编程、开发角色轮替、集成开发环境、敏捷开发（极限编程、增量开发）、设计模式、可视建模等软件开发的内容，取得了较好的成效。但此方面的任何一个内容也十分庞大，如何在学时有限的条件下合理应用成熟的软件工程方法及实践需要结合教学内容进行凝练。

随着多核、众核、并行计算及大数据（Big Data）技术的飞速发展和普及，在第一门计算机课程里是否引入与之相关的内容是值得考虑的。除了相关的基本概念及基本方法之外，如何对进程调度、异步事件驱动处理、分布式存储和大数据的处理架构等关键技术进行深入浅出的讲授及合理安排训练，对教师是个巨大的挑战。

在编程训练中通过浏览器配合在线学习及编译的网站进行教学及实验是简化编程训练的一个重要途径。常见在线编程网站有 Treehouse、Codecademy、LinuxCast、LearnStreet、Code.org、

Topcoder、Codeevy、Codeschool 和 Tryhaskell.org。另外随着云计算技术的普及也有很多面向云计算的开发工具出现，为编程训练提供了新途径，常见工具包括 Cloud9 IDE、Codeanywhere、Coderun、Cloud IDE、Kodingen、ShiftEdit、Akshell、Orion 和 CodeTester。

六、结论

对比中美高校第一门计算机课程，我们在对计算机思维关注程度、课程设计思路及课程模式等宏观方面与美国差距不大，但是在具体课程的微观方面还有不少差距。美方课程的开设为我们提供了较宽的视野、较新的课程内容和较好的教学方式。各个学校应该根据自己学校的定位、学生特点和专业要求，按照“分类分层次”的原则，结合计算思维能力培养，在教学内容及教学方法方面进行深入研究，在理解美国高校课程的设计思路基础之上选取合适的内容及方式，对我们的第一门计算机课程进行改革和实践。

参考文献：

[1] R. E. Bryant. Introductory Computer Science Education at Carnegie Mellon University: A Dean's Perspective[R]. Technical report CMU-CS-10-140.10.1007/s11390-011-1157-0.

[2] 教育部高等学校计算机基础课程教学指导委员会. 高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求[M]. 北京：高等教育出版社，2009.

[3] 李廉. 计算思维——概念与挑战[J]. 中国大学教学，2012(1).

[4] 李波. 计算思维与大学计算机基础[J]. 中国大学教学，2012(7).

[5] The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society[Z]. Computer Science Curricula 2001, February 2001

[6] The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society[Z]. Computer Science Curricula 2013. February 2013

[致谢：本文的撰写是集体智慧的结晶，在陈国良院士的领导下参与工作的同志有张铭教授、董荣胜教授、张龙副编审等]

[责任编辑：余大品]

（上接第 85 页）经费如何支付给企业也不清楚。有学校指出，到外地开展实践教学的经费缺口较大。

（5）课程体系重构遇到困难。“卓越计划”强调工程实践能力、工程设计能力与工程创新能力的培养，这就需要重构新的课程体系。但长期以来，自动化专业课程设置过分强调学科的系统性、逻辑性，按学科设置课程，忽视学科之间的整合、关联。现在重构新的课程体系，部分教师会不适应。

（6）学生对“卓越计划”的培养目标和要求需要进一步加深认识、统一思想。少数学生以考研为目的，希望在低年级时享受“卓越班”浓厚的学习氛围，在高年级时不进入企业实习，退出“卓越班”专心复习考研。

（7）需要加强“卓越计划”实施过程的管理，特别是学生在企业学习阶段的安全、企业技术保密等方面的管理。

五、有关建议

为了保证“卓越计划”的顺利实施，各校提出了多项建议与意见。主要有以下几个方面：加大实践教学条件建设，加大教师工程能力培养；加强对“卓越计划”试点专业的经费支持；调动企业培养人才的积极性，让企业乐于加入到培养人才的行列中；对研究型大学的“卓越计划”专业点，建议在专业硕士研究生名额上给予政策倾斜；逐步制定和完善卓越工程师教育培养标准，等等。

[责任编辑：夏鲁惠]