计算机学科 教学方案 计算机概论 (6)

计算机科学 1995 Vol. 22 No. 1

21-26 TP301

计算机学科教学方案(91)剖析

严蔚永

(中国纺织大学计算机科学与工程系 上海 200051)

A 摘 要 本文讨论和分析了 ACM 和 IEEE-CS 计算机学科教学方案 1991，着重介绍其原则、思想和观点，旨在促进我国计算机学科之发展。

一、引言

美国计算机协会(ACM)和 IEEE 的计算机学会(CS)于 1985 年成立一个联合任务组(称为 Task Force)，由各方面的著名计算机专家学者组成，旨在研究并提出一个新的计算机学科教学方案。1991 年 3 月发表了这个方案，称为"COMPUTING CURRICULA 1991"[5]。本文译为"计算机学科教学方案 1991"，以下简称"方案 91"。

ACM 和 IEEE-CS 都提出过计算机科学与工程的教学方案与计划[1,2,3]。方案 91 继承了先前的工作，又作了发展和改进，考虑了近年来计算机学科所取得的巨大发展与进步。先前的工作对方案 91 的影响主要有如下几点：

1. 联合任务组于 1988 年 12 月发表报告[4]，提出了对计算机学科(Computing as a discipline)的全面定义。该报告(下称"报告 88")强调实验教学与课堂教学的结合，肯定培养学生设计能力的重要性，并建议基干课程(称为"引导性课程")应遵循广度优先的原则。

2. IEEE-CS 方案 83[3]对各个重要课目进行了深入说明，包括实验的说明，并采用模块化方法构成课程。

3. ACM 方案 78[2]对各课程也给以详细说明，并确定程序设计的突出作用。

方案 91 则进一步明确提出"理论"、"抽象"、"设计"三种进程(process)，它们贯穿于九个"主题"领域，并强调十二个所谓"复发概念"。

二、本科教学计划的目标轮廓

联合任务组认为：大学本科教学应使学生既在学科的学术方面又在社会所需的职业技能方面均获得培养。毕业生应知道计算机学科的历史，包括其主要发展和趋势，及在经济、科学、法律、政治和文化方面的表现；这些历史进程的组合，在较短暂时期内形成了这一学科。方案 91 认为，在本科教学计划的目标方面应考虑下列几点。

1. 提供连贯协调和宽广的计算机学科知识，使学生在相当程度上掌握各主题及其进程，以及相互关系。

2. 在宽广的知识结构框架内有效地发挥各院校的特色，以使学生获得有效的训练。各院校可依据自身的条件和特长有不同的侧重。有的可注重培养的广度、有的可注重培养的深度；有的在各方面要求之间保持严格的平衡，而有的可灵活一些。

3. 教学计划可在目标方面予以不同程度的加强，即在培养学生的职业技能，使学生能继续攻读研究生学位，或使学生能迎接更为广泛的职业和个人生活的挑战等方面可以有不同程度的侧重。

4. 提供一种环境，使学生能接触到与计算机学科领域有关的论理和社会问题。这种环境应与当前技术与理论的发展相适应，并坚持一般的职业规范，使学生既认识到自己的能力与局限性，又认识到学科本身的能力与局限性。

5. 使学生能充分掌握计算机学科的丰富理论，以致能对一些较为深刻和抽象的论题进行评估和研讨。

三、计算机学科的定义

报告 88[4]给出了计算机学科的全面定义。这个定义给教学方案与计划的制定提供了一个概念基础。

报告 88 指出，计算机科学与工程是对描述和变换信息的算法过程(包括它们的理论、分析、设计、效率、实现和应用)进行系统化的研究。计算机学科的基本问题是：在算法过程中什么能够(高效率地)自

严蔚永 教授

动化?

为了阐明学科定义,简要回顾一下学科发展的历史.计算机学科的发生可以追溯到四十年代早期,由于算法理论、数理逻辑、及存储程序电子计算机的发明这些方面的结合,产生了这个学科.计算机学科探探地扎根于数学与工程.数学指导分析,工程指导设计.与大多数物理学科不同,计算机学科既包含着理论和试验,又有工程技术,而物理学科与应用其原理的工程学科是不相同的.然而在计算机学科中科学与工程是不可分离的.

在人类文明历史发展的数千年中,演算(calculation)一直是数学的主题.人们建立了许多物理现象的模型,导得数学方程,求得其解,从而对现象给以本质的揭示,预示发展的进程,比如,天体轨道、气象预报、流体流动等的数学推导,都为此例.为求解方程,人们创造许多通用方法,比如线性方程组、微分方程、积分函数等的算法.几乎在同一个时期,机械系统设计中的演算也一直是工程学科的主题,比如,静止物体中应力计算、运动物体动量计算、宏观微观距离测量.

这样,工程与数学的长期相互作用的一个必然产物是演算的机械辅助工具:从远古时代的石块、绳结,到后来的各种各样的算盘,直到现代计算机.中国古代出现过许多伟大的数学家和发明家,对数学与演算工具作出过巨大贡献.十七世纪中叶帕斯卡和莱布尼兹造出了算术演算器.十九世纪三十年代Babbage设想了一种解释器,能机械地计算对数、三角函数、及其他通用数学函数.他的机器虽未完成,但给后来的工作提供了启示.在本世纪二十年代Bush制造了电子模拟计算机,用以求解一般微分方程组.同期,也发明了电子机械演算机器,用以进行四则运算和平方根计算.电子触发器的发明,使这类机器过渡到数字电子形式成为可能.

另一方面,逻辑是数学的一个分支,它研究推理(Inference)成立的判据和推论(Reasoning)的形式原则.自从欧几里得时代以来,它一直是严格进行数学和科学论证的工具.十九世纪,有些人开始探索一种万能的逻辑系统,并设想这种系统没有在已知的演释系统中所观察到的那种不完备性.假如有这么一个完备系统,那么就可能机械地确定任何一个给定的断言是真还是假.然而1931年恪德尔发表了著名的不完备性定理,证明这种系统是不存在的.后来,在三十年代,图灵提出了通用计算机的思想,它可能模仿任何其他计算机器的任何一步步的步骤(procedure).图灵的发现与恪德尔的发现相类似:某些明确定义的问题是不能用任何机械的步骤来解答的.逻辑的重要性不仅在于它深刻揭示了自动化演算的限制,而且还在于揭示符号序列(可能编码或数)既可作为数据来解释,也可作为程序来解释.后者乃是存储程序计算机与演算机器不同的关键.算法步骤可编码成机器表示,并存在存储器中,以便由处理机对它进行解码与执行.这种机器代码可机械地从一种高级符号形式(程序设计语言)推导得来.

长期的数学演算与逻辑符号处理作为一条发展线索,与另一条信息的电子表示的发展线索相互综合在一起,促使了计算机学科的诞生.为此,报告88提出了九个子学科(即主题)、三种处理方法学(即进程),并给出具体说明.

四、方案的设计原则

1.九个主题

报告88将整个学科范围划分为九个子领域,称为主题,复盖整个学科.每个主题都包含相当份量的理论、抽象、设计以及实现等方面的内涵.但是,并非这些主题中每个课题都是各种教学计划所共同要求的,其中有些课题可以作为本科高级课程或研究生课程的内容.方案91采用了这九个主题的划分.它们是:(1)算法和数据结构(AL);(2)系统结构(AR);(3)人工智能和机器人学(AI);(4)数据库和信息检索(DB);(5)人机通信(HU);(6)数值和符号计算(NU);(7)操作系统(OS);(8)程序设计语言(PL);(9)软件方法学与软件工程(SE).报告88和方案91给出了这些主题的详细说明.

2.三种进程

计算机学科是数学、科学和工程相结合的学科.在不同的学科中,人们采用不同的方法学处理问题.报告88和方案91分为三种方法学,称为进程.

第一种进程称为"理论".这是一般数学所采用的方法,用于发展严密一致的数学理论.方法可归纳为四步:

1)刻画研究对象的特征(给出定义);

2)就这些对象之间可能的关系提出假设(提出定理);

3)确定这些关系是真或假(定理证明);

4)解释所得结果.

当发现错误或不一致性时可以重复这些步骤.学生进入大学以后,首先可能在数学课程中接触到这种进程.后来在诸如算法的复杂性理论、逻辑、及形式语言与自动机这类课题中也会用到这种方法.方案91包含了相当份量的理论.

第二种进程称为"抽象".这种方法也可称为"建模"(Modeling).这是实验科学所用的方法,用于对

客观现象的研究。方法也可归纳为四步：

1)由对现象的观察形成假说(提出假说)；

2)构造模型，分析模型，做出推测(建立模型)；

3)设计试验，做试验并搜集数据(试验)；

4)分析所得结果(分析)。

当从模型所得的推测同试验结果不符合时可重复这些步骤，即修改或重新形成假说，构造模型及做出新的推测。这一进程既可在课堂教学也可在实验教学中实施。比如，冯·诺依曼计算机模型就是一种抽象，可以将它的特征进行分析并与其他模型作比较。学生在低年级的课程中就可以接触到这种模型抽象。在实验室试验工作中也可以进行对抽象的分析，如研究计算的限制、新的计算模型、以及各种尚未证明的理论猜想。

第三种进程称为"设计"。这是工程技术的方法，用于研制一种系统或设备，以解决某个给定问题。方法也可归纳为四步：

1)分析和陈述需求(需求分析)；

2)分析和陈述规格(制定规格)；

3)设计和实现系统(设计实现)；

4)测试所实现的系统(测试)。

这些步骤也可以重复。学生可以通过他人的设计及自己的实践来学习设计。教学中可以进行许多面向设计的实验室大作业，以使学生获得设计的第一手经验。这些大作业着重培养学生综合解决实际问题、学会在实际条件下评估各种方案的性能和费用、以及作出抉择的能力。

每个主题都存在这三种进程，有时紧密交织在一起。三者反映对学生三种能力的要求。理论涉及描述和证明对象间关系的能力；抽象涉及运用这些关系做出推测并同实际情况作出比较的能力；设计涉及实现这些关系的具体实例以及运用它们实施有用动作的能力。通过训练学生应掌握这三种能力，并达到相当的强度。比如，在设计方面，运用先进的 VLSI 设计和仿真系统使学生能掌握微电子技术的高效与正确的设计能力；又如运用软件开发环境使学生能掌握软件的高效设计能力。在抽象方面，使学生掌握运用功能强大的计算机对实际问题进行建模和推测的能力。

3. 十二个复发概念

任务组认为，在计算机学科中经常出现并在制定各门课程和整个教学计划中起重要作用的有十二个最基本的概念，称为复发概念。这些概念是把整个学科统一起来的一些重要思想和原则。能否融会贯通这些概念并把它们应用于具体场合，这是本学科大学毕业生是否成熟成为计算机科学家和工程师的一个标志。显然，在制定具体教学计划中，这些概念必须有效地互相配合和衔接。恰当地运用这些概念是方案 91 具体实施的基本点。这些概念也可作为将教学内容组织起来构成具有内在有机联系的各门课程的基本线索。这些概念有三个基本特点：贯穿于整个学科；有各种各样具体实例；有高度技术独立性。这些概念渗透到各方面，与具体技术相对无关，比具体实例更为基本，在学科成长历史中一直起着作用，在可见的将来也会继续起作用。下面简单介绍这些概念。

1)束定(binding)：给抽象的事物添加一些属性使之变为较为具体的事物这样一种过程。例如，将一个进程指派给一个处理机；数据类型结合于具体变量名；子程序的调用引导到库目标程序的连接与执行；逻辑程序设计中的实例化；面向对象程序设计中方法结合于消息；根据抽象的描述创建具体实例；这些都反映"束定"这个概念。

2)大问题复杂性(complexity of large problems)：随着问题规模的增长其复杂性非线性增长效应。在不同的数据规模、问题范围和程序大小的条件下，在区分和选择不同的方法时，这是一个重要的考虑因素。其实，计算机系统(无论是硬件还是软件)的奇妙特点也产生于巨量简单性所形成的复杂性，问题的困难和障碍也往往在于处理这种复杂性。例如，计算机的单步操作可以非常简单，而许许多多这种单步连贯起来却能解决非常复杂的智能问题；计算机的单元组成可以是非常简单的 0/1 门，而千百万个门组成的集成电路却构成非常复杂的功能行为。有时要利用这种复杂性，有时要减少这种复杂性。复杂性在实践中往往用时间或空间测度来衡量。对于同样复杂性的问题，力求降低解决此问题所用的算法复杂性；对于同样复杂性的算法，力求去解决更为复杂的问题。

3)概念模型和形式模型(conceptual and formal models)：对一种思想和问题加以形式化、特征化、可视化以及进行抽象思考的各种方法。例如，逻辑、开关理论、计算理论中的形式化模型；程序设计语言的形式化模型，语法形式规范(paradigm)；用于规范和设计系统的可视化语言，如数据流图(DFG)和实体关系图(ER 和 ELR)。

4)一致性和完备性(consistency and completeness)：计算机学科中，一致性(无矛盾性)和完备性的具体体现，包括诸如正确性、健壮性(鲁棒性)、可靠性、完整性这些相关概念。例如，作为形式规格的一个公理集合的一致性；理论相对于被观察事物的一致性；一个语言或一个接口设计的内在一致性，正

确性被视为部件的或系统的行为是否符合规格要求的这种一致性。完备性是指如公理集合是否抓住全体所要求行为的这种充分性；软件和硬件系统功能上的充分性。系统在错误条件和未预计情况下仍具有良好行为的能力，这被视为健壮性，这也是一种完备性。可靠性也是完备性的一种实际测度。完整性是指系统始终保持一致性的一种机制和能力，防止授权用户不正确访问系统的能力。

5)效率(efficiency)：相对于资源的费用测度。资源包括时间、空间、存储、人员、甚至金钱等。例如，算法的空间与时间复杂度的理论估计；实现某个所要求结果(如完成一项计划，生产一个产品)的费用；各种可选实现方案的比较，等等。

6)进化(evolution)：事实上的变化及其影响。这包括变化在各种程序上引起的冲击；抽象、技术和系统在面对变化的形势中所具有的恢复力和充分性。例如，一种形式模型在历史进程中仍能有效表达系统行为的能力；一个设计在环境、需求、实现手段和管理方法改变的情况下仍能保持有效性的能力。由于计算机技术的飞速发展，进化及适应进化的能力问题日益受到广泛的重视。兼容性问题是进化的实际体现。开放系统概念是适应工业实践中的进化而提出来的。进化即演变，演变产生差异，在差异中抓住不变性，这是开放系统生存和发展的关键。开放系统具体反映在四个方面，即可移植性、互操作性、连接性和规模可伸缩性。

7)抽象级(levels of abstraction)：指抽象所具有细节和具体特性的程度。在复杂度控制、系统结构化、细节隐藏、以及抓住反复出现的本质属性方面，用抽象级概念来刻画抽象的程度。例如，硬件描述的层次；虚拟机的概念；在分层结构的对象中规定性的等级；程序设计语言中的类属概念；在问题用程序求解过程中细节被提供的程度；抽象程序概念。

8)空间有序性(ordering in space)：在空间中事物的次序。计算机学科中的局部性和邻近性概念是空间有序性的直接延伸。空间中位置的概念是空间有序性的基本特性。在存储器中和网络配置中有物理上的位置概念，这是欧氏空间位置概念的物理实例。此外，还有组织上的位置概念，如处理机、进程、类型定义及有关操作，这些成分代表一个组织空间中的不同位置。还有逻辑上的位置概念，如软件作用域、模块间的耦合性和模块内聚性，这些代表抽象空间中位置。有序性概念建立在空间(更抽象为数学中度量空间概念)中距离概念的基础上。

9)时间有序性(ordering in time)：事体发生在时间上的次序。在形式模型中用抽象时间作为参数；在时态逻辑中的时间参数，这些都是实例。时间作为空间上分开的不同进程之间的同步手段。时间也是算法执行中一个基本要素。

10)重用(reuse)：一个特定技术、概念或系统成分在新的场合和情况下再次被运用的能力。比如，可移植性；软件库重复使用；硬件成分重复使用；用于开发可重用软件模块的语言抽象机制。资源共享是计算机学科中一个基本问题，包括代码共享和数据共享。代码的可重入性保证代码共享性。

11)安全性(security)：软件和硬件系统适当地响应和抵御不适当的、未预计或未授权请求的能力。比如，计算机装置抵抗灾难性事件(如自然灾难、有意破坏)的能力。又如，类型检查以及为防止数据对象和功能被误用而设置的语言概念；数据加密；数据库管理系统中特权的授予和使用；保证用户最少出错的用户接口；计算机设备上的物理安全设施；系统中各种等级的安全机制；数据访问控制；抗病毒机制。

12)权衡及其后果(trade-off and consequence)：系指计算机学科中的权衡现象及其引起的后果。权衡选择是各子学科一个基本事实。为达到某个目标一般存在多种可选途径，尤其在工程技术领域是如此。例如，各种设计方案选取中的技术、经济、文化和其他效果的比较分析；研究算法中空间时间权衡；设计中互相冲突的目标之间的权衡；硬件与软件之间权衡。

在根据方案具体制定教学计划时，制定者必须了解这些复发概念的基本作用，也就是说，一个复发概念(或一组复发概念)有利于将一门课程、一次讲授和一个实验作业系统地组织起来，否则它们很可能显得杂乱无章，无从把握。对于整个教学计划，也是如此。这些概念是一条或几条线索，把各门不同的课程编织在一起。通过这些概念反复出现，使学生理解整个计算机学科是一个具有内在有机统一性的学科。

4. 社会与职业环境的有关问题

学生需懂得与计算机学科有关的文化、社会、法律及伦理方面的基本问题。应当了解计算机学科的过去、现在和将来；了解个人在这一历史进程中的作用；对学科发展中起着重要作用的哲学问题、技术问题、美学问题等能够做出评估分析。学生应能对学科发展的社会效果提出自己的看法。学生应了解有关的法律权利和义务、以及伦理基础；比如，了解知识产权的法律权利和义务，及其在计算机软件产品上的解释。方案91提出了相应的知识单元"社会、伦理和职业问题"(SP)。

五、如何将原则贯彻于方案

整个教学过程应使学生在各主题领域都达到相当的水平，并达到足够的广度，而不是仅限于局部领域。方案提出了一组共同要求以保证这种广度的实现。共同要求还提出了适当的深度要求，且不同的具体教学计划可以有所不同。方案中关于"高等补充课程"的建议进一步保证深度的实现。方案91认为，在理论和设计这两种进程方面，不同教学计划可有所不同的侧重，即有的偏重于理科，有的偏重于工科；但抽象进程则都应注重。共同要求对三种进程作了规定，并结合实验室教学以加强有关课题。图1为原则如何贯彻于方案的示意图。其中"其他等级要求"系指各院校的一些特殊考虑。下面扼要介绍方案的若干基本观点。

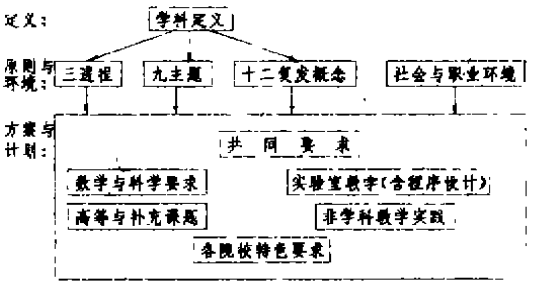

图1 原则与方案

1. 广度与深度的合理结合

广度要求是基本的，特殊的深度要求是必需的补充。

为了保证广度，在制定教学计划时应提供九个主题领域的广泛的选课范围。方案没有提出单一的课程设置，各院校可自行其事，只要达到共同要求。共同要求包含许多知识单元，可以不同方式组合成不同的课程设置。方案给出了一些实例。

为了保证深度，不同院校可根据其教学使命、教师队伍状况和学科方向提出进一步的要求和选择。方案91提出许多高等与补充课题以供选取。

由此可见，方案91没有限制各院校的手脚，既规定原则和共同要求，又给予各院校以充分的独立自主权和灵活性，以有利于创造多样性的教学计划，有利于人材培养和学科发展。

2. 程序设计的恰当地位

程序设计是计算机学科贯穿各子学科的基本活动和方法，必须给以恰当的地位。程序设计不能仅理解为具体的编程，而应包含更深更广的内容，包含掌握算法基础、技巧与风格、语法与语义学、语用概念、语言加工处理、编译与翻译等。熟练掌握程序设计是通过各种途径实现的。应尽量提早让学生懂得程序设计，并在整个教学过程中逐步加深和熟练掌握。报告88曾提出"计算机科学就是程序设计"，这并不恰当，虽然后者非常重要，但前者内涵更广。

3. 实验教学的重要性

报告88对实验教学的目的和结构作了说明，方案91进一步作了规定。认为讲授主要解决原理问题，而实验解决技术问题。实验教学的主要目标是：

· 实验应当验证讲授中所包含的原理如何被用于设计、实现和测试实际的软硬件系统。学生通过实验获得具体的经验，以帮助他们理解抽象的概念。这些经验不仅对于加深学生关于计算机学科的直觉认识是重要的，而且对于在构造正确高效计算机程序和系统中加强学生的智力开发也是重要的。

· 实验应当着重于引导学生获得对计算机学科实现技术的深刻认识。实验应强调程序设计而不是程序本身；强调实验方法和技术；强调对硬件功能的理解；强调正确运用软件工具与文档，以及正确地掌握编制文档的能力。在实验所用的宿主计算机上应当配置充分的软件工具，以支持各分系统或工作站上进行实验。实验应使学生掌握工具的正确使用。

· 实验应当使学生掌握试验方法学，包括进行试验及设计试验；使用软硬件监测工具及设计监测；试验结果统计分析及报告；学会将本质现象与偶发现象区分开来。

方案强调实验教学的重要性，以及建立良好实验手段和设施的重要性。实验条件必须跟上技术的发展，保持先进性。

方案91在每个知识单元中都包含若干实验项目。实验有两种类型：开放性实验和封闭性实验，开放性实验无教师指导，由学生自己制定计划加以完成。常规程序设计可以安排在开放性实验中。封闭性实验有教师指导，按规定要求和时间完成。

4. 注意非学科教学实践的锻炼

除了学科本身以外，本科学生还应当获得其他方面的教学实践经验，这些经验对他们在思维能力、问题求解、研究方法和职业技能等方面的培养也是十分有利的。这些实践可以包含在课堂讲授、实验和课外活动之中，比如：参加集体群组活动，培养协作能力；同他人进行口头或书面交流，培养交流信息能力；参与各种社会实践，熟悉有关社会职业。

六、共同要求

方案包含了一系列共同的要求，即各具体教学计划必须达到的要求，保证计划的广度和深度。共同

要求形成方案91的基础，也构成各种具体教学计划的知识主体，共同要求用知识单元的形式表示，而不是以课程的形式表示，以便于不同院校和教学计划以不同方式组合课程，这提高了方案的灵活适应性，也充分调动了院校和教师的积极性。

1. 共同要求的基本构件：知识单元

共同要求按九个主题，外加两个可选题的次序组织起来。这两个可选题是"程序设计语言导引(PR)"和"社会、伦理和职业论题(SP)"。每个主题和可选题由若干知识单元组成，这可视为共同要求的"细胞"。每个知识单元以主题或可选题的缩写代码后加数字加以标记，如AL1，DB2等等。

图2 为每个知识单元的描述格式。

```
〈缩写代号〉:〈知识单元名称〉
〈内容概要〉
复发概念:〈概念1〉,〈概念2〉,……
讲授课题:〈〈时数要求〉〉
〈课题列表〉
建议实验:〈〈实验类型〉〉
〈实验内容〉
衔接:
  关连单元:〈缩写代号列表〉
  前提单元:〈缩写代号列表〉
  后续单元:〈缩写代号列表〉
```

图2 知识单元描述格式

2. 高等与补充课题

一个完整的方案不仅包含共同要求，还应包含一些附加的要求。高等与补充课题使学生获得更深的学科知识。附加课程的数量因院校而异。从建议的内容看，这些课题包含一些较新较深入的题目，如高等操作系统、高等软件工程、计算机图形学、计算机网络、并行与分布计算、实时系统，等等。显然，这是变化发展较快的部分。

3. 数学与自然科学要求

方案91对数学和自然科学要求作了单独的规定，并给以相当的份量。数学包括离散数学、微积分、数理逻辑及各门应用数学。自然科学对计算机学科十分重要，尤其是物理学和生命科学。

七、具体教学计划的制定

具体教学计划的制定取决于许多因素，如计划的目的、教师队伍的力量、学生的背景和志愿、教学支持资源、基础结构(组织机构、人员编制、基金资助等)，以及其他有关原则。

教学计划制定应遵循下列原则：

(1)计划安排应给学生提供途径，以获得学科规定知识、专业技能和经验，毕业时达到应有水平。

(2)整个计划应贯彻统一的思想和目标，如抓住复发概念作为统一思想的依据。

(3)应提供使学生获得额外的专业知识和技能的多种途径，包括课外教学机会。

(4)要考虑计划实施的现实约束条件。

(5)要考虑计算机学科的新发展，使学生适应这种情况。

在制定计划时，方案建议应考虑计算机学科的飞速发展，定期对计划进行评估和修正。计算机学科理论本身与其他一些自然科学理论不同，发展变化也比较快。计算机科学是建立在人造事物基础上的，不仅要服从一般科学定律，而且其基础理论也要不断重新解释和重新评估。

课程设置依据共同要求来设计，由知识单元组成课程。所有知识单元规定学生应掌握的学科范围，每个知识单元的授课时数反映了该单元的大致深度。相关的单元表明课程需某种顺序。

方案91最后介绍了几种具体计划，可作为制定者参考。

八、结束语

本文对ACM和IEEE-CS计算机学科教学方案1991进行了讨论和分析，着重介绍其原则、思想和观点，具体教学内容没有赘述，读者可参考原文。方案91对我国计算机专业的教学很有参考价值，可加以吸收；但也不应照搬，应考虑到我国的实际情况和条件。总之，为使我国计算机专业教学与国际水平相适应，同时结合我国实情，我们应当很好地研究方案91。

参考文献

[1]ACM Curriculum Committee on Computer Science. Curriculum 68: Recommendations for the undergraduate program in computer science, 《Commun. ACM》11, 3(Mar. 1968), 151-197

[2]ACM Curriculum Committee on Computer Science. Curriculum 78: Recommendations for the undergraduate program in computer science, 《Commun. ACM》22, 3(Mar. 1979), 147-166

[3]Educational Activites Board. The 1983 Model program in Computer Science and Engineering, No. 9 2, Dec. 1983

[4]Denning, P. J. et al. , Computing as a discipline, 《Commun. ACM》32, 1(Jan. 1989), 9-23

[5]ACM, IEEE-CS Joint Curriculum Task Force, 《Computing Curricula 1991》, Feb. 1991