

第三讲

计算机学科专业、 课程与知识体系

南京大学计算机科学与技术系

黄宜华

yhuang@nju.edu.cn

了解本学科体系的重要作用和作用

ACM/IEEE在2005计算机课程报告中明确指出

- 计算机分支学科和内容不断增加，试图精通本学科的所有方面对任何人来说都是一件不可能的事情；
- 计算机分支学科的增加要求我们明确清晰地描述出各分支学科所具有的共性特征，以及各自不同的特征，以便帮助学生、教师及社区进行恰当的专业规划和选择；
- 我们必须在本科层面上，从每个分支学科的知识体系和技能的角度，清晰明确地描述每个分支学科。

了解本学科体系的重要作用和作用

作用

- 帮助同学们了解本学科的整体概貌和特征
- 帮助同学们了解本学科下的专业领域，以便同学们能根据自己兴趣特长和未来的职业发展规划进行恰当的课程规划和选择
- 帮助同学们了解本学科下的课程和知识体系，以便同学们了解相应的学习内容、学习方法和学习目标

什么是计算机学科？

- 计算学科、计算机学科 Computing Discipline
计算机科学与工程 Computer Science & Engineering
计算机科学与技术 Computer Science & Technology
- 笼统定义：计算学科是一种需要使用、从中受益、或者创建计算机的一门学科 In a general way, we can define computing to mean any goal-oriented activity requiring, benefiting from, or creating computers.
- 综合性学科：计算机学科是一个广泛涉及到如数学、物理、电子工程、信息学、语言学、认知科学、艺术、哲学、心理学等基础性学科的综合学科。

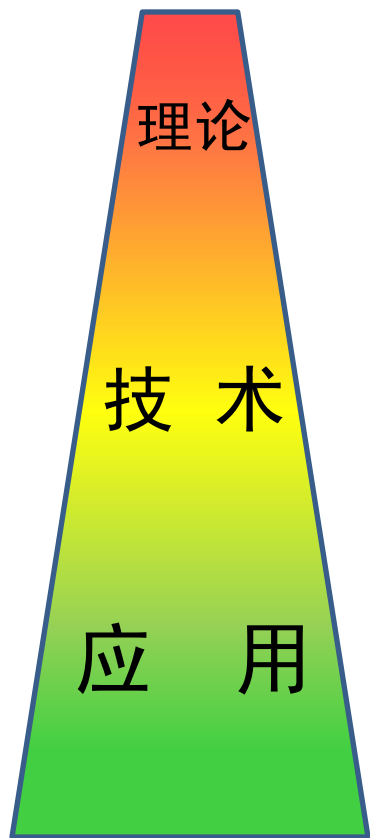
难以穷举的计算机学科内容

- 设计与构建计算机硬件和软件系统 designing and building hardware and software systems for a wide range of purposes.
- 处理、组织和管理各种各样的信息, processing, structuring, and managing various kinds of information.
- 进行科学研究, Doing scientific studies using computers.
- 使计算机系统具有智能making computer systems behave intelligently.
- 制作与使用通信媒体和娱乐媒体, creating and using communications and entertainment media.
- 为某个特定目的查找与收集相关信息, finding and gathering information relevant to any particular purpose.
- 更多.....

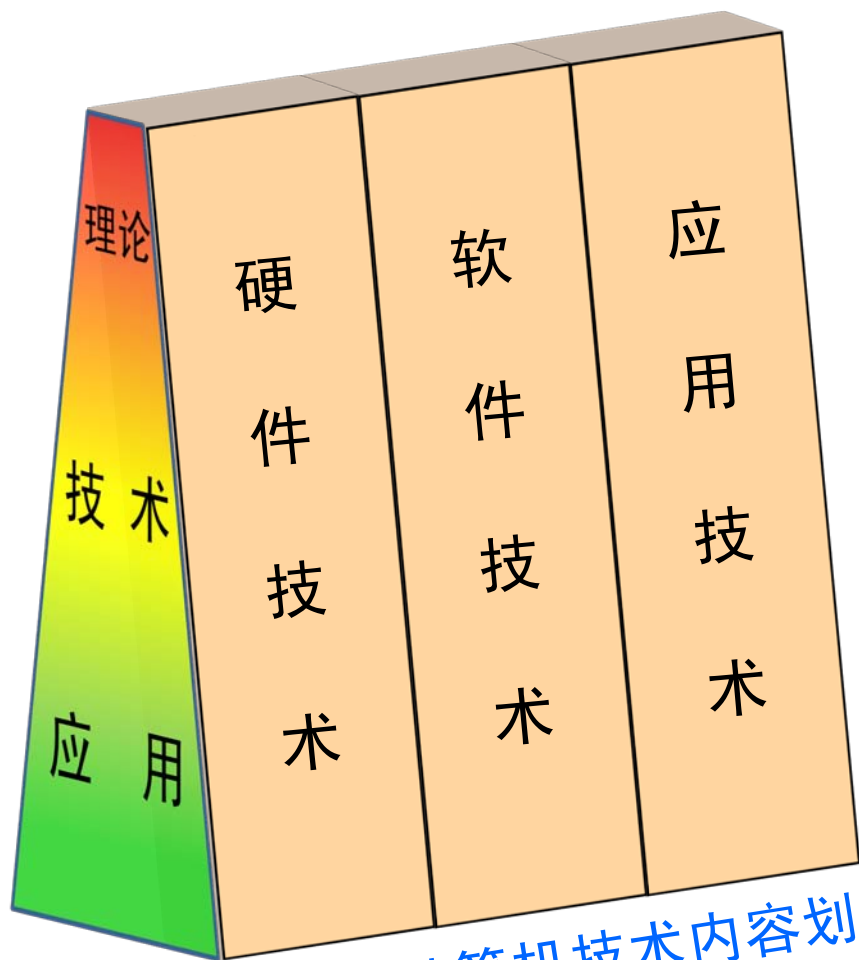
计算机学科的特点

- **基础性：** 需要掌握一些数学理论基础，以及计算机专业的基础理论、原理和技术方法。
- **实践和应用性：** 把发现新的科学知识与应用这些知识去解决实际问题紧密结合，理论上或者方法上的创新往往很快就在产品和应用中得到体现。
- **工程性：** 不仅研究计算机硬件和软件的原理，更侧重于它们的实现，即如何将计算机硬件和软件具体构造出来。
- **多样性：** 知识面宽，宽度大，它与数学、电子、通信、工程、管理等多种学科相关。
- **发展变化的快速性：** 知识和技术发展变化速度快。
- **就业方面的优点：**
 - 就业面比较宽，毕业生在许多行业里都可以找到合适的工作；
 - 在诸多专业里，计算机专业人员的平均起步收入应该是很高的；
 - 具有挑战性和创造性，容易找到自己喜欢的领域和岗位。

计算机学科的观察角度



一般学科内容性质划分



传统的计算机技术内容划分

教育部规定的计算机分支学科

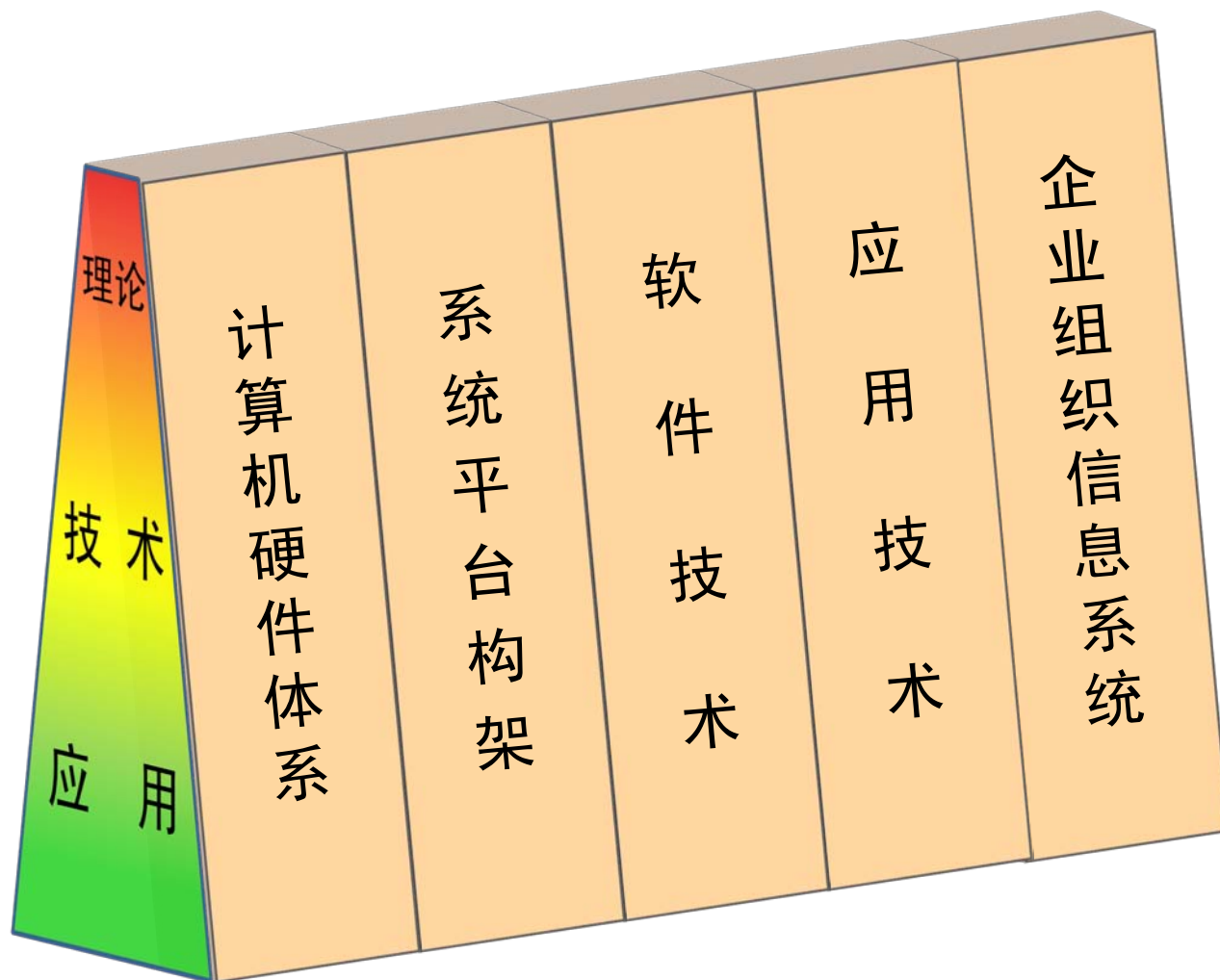
教育部本科专业目录规定，大学计算机系本科学士生一般不再划分专业(?)，以打好基础为主。

研究生教育阶段，计算机学科分为3个二级学科(专业)

- 计算机系统结构专业
- 计算机软件与理论专业
- 计算机应用技术专业

ACM/IEEE计算机分支学科

ACM/IEEE CC2005计算机技术内容划分



ACM/IEEE计算机分支学科

CC2005建议的主要分支学科划分

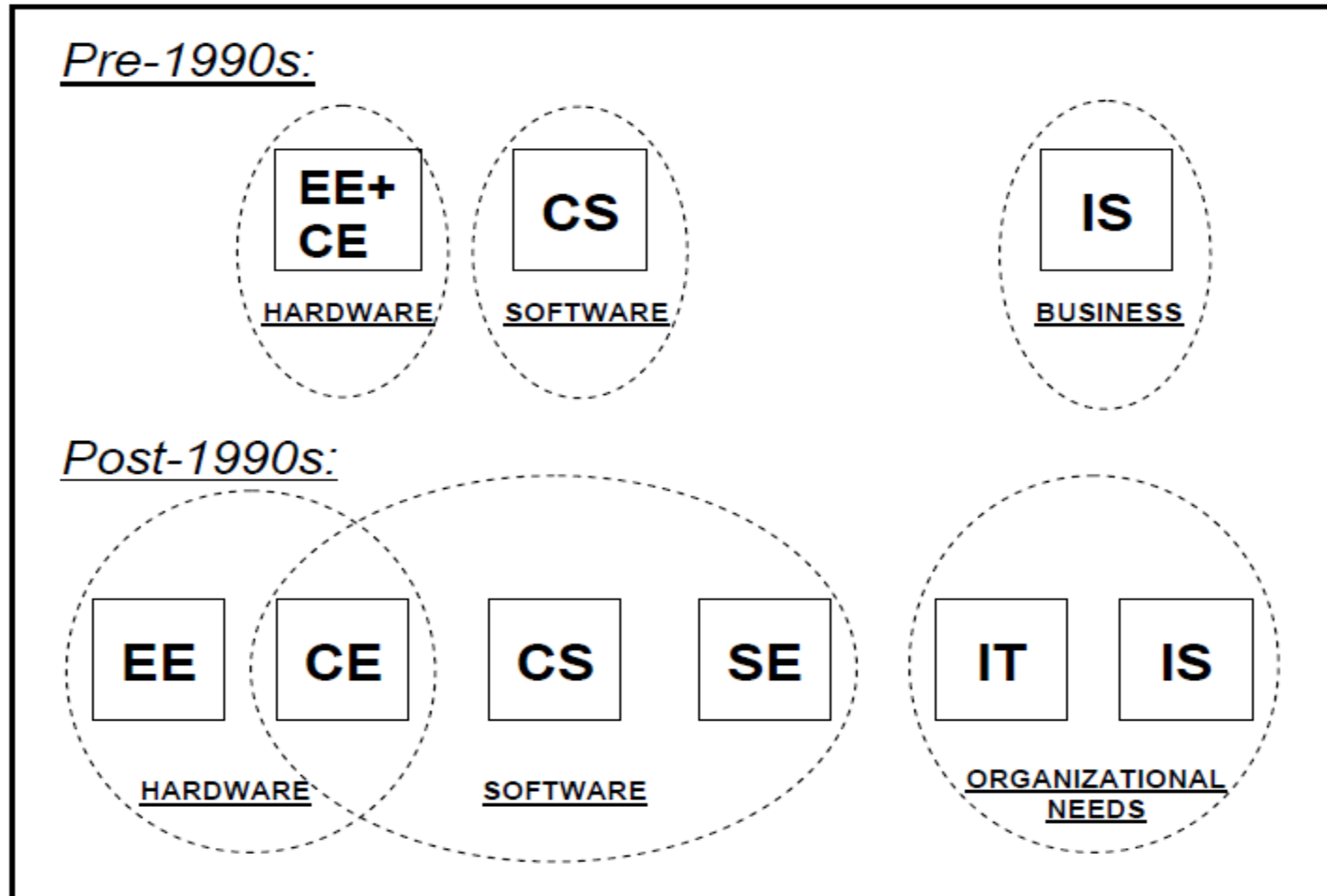


Figure 2.1. Harder Choices: How the Disciplines Might Appear to Prospective Students

ACM/IEEE计算机分支学科

CC2005建议的主要计算机分支学科

- 计算机科学 (Computer Science, CS)
- 计算机工程 (Computer Engineering, CE)
- 软件工程 (Software Engineering, SE)
- 信息系统 (Information System, IS)
- 信息技术 (Information Technology, IT)

密切相关的基础或热点学科

- 数学等基础学科
- 电子工程 (Electrical Engineering, EE)

主要计算机分支学科

教育部计算机学科教学指导委员会

2005年在进行广泛调查研究的基础上建议，我国本科计算机专业应以“培养规格分类”，不同学校根据社会的需求和自身情况，为学生提供如下3种类型(4个专业)的教学计划和培养方案：

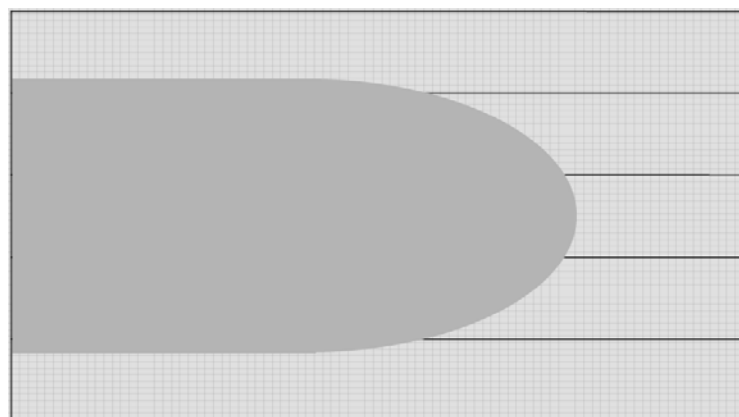
- **研究型** (与CC2005的“计算机科学”专业对应)
- **工程型** (与CC2005的“计算机工程”专业和“软件工程”专业对应)
- **应用型** (与CC2005的“信息技术”专业对应)

ACM/IEEE计算机分支学科

计算机科学(Computer Science, CS)

关于计算机学科的理论、算法基础,到最前沿的如机器人、计算机视觉、智能系统、生物信息学等应用领域的理论和基础研究;着重于信息和计算的理论基础问题研究,也包括计算机系统实现和应用中的基础理论和方法研究。

企业组织和信息系统
应用技术
软件技术
系统平台构架
计算机硬件体系



理论
原理
创新

理论

技术
工程
系统

技术

应用
部署
配置

应用



ACM/IEEE计算机分支学科

计算机科学(Computer Science, CS)

主要研究领域包括:

- **理论计算科学**: 经典的计算理论以及其它一些关于计算技术中较为抽象、逻辑和数学化方面的主题。
 - **可计算理论**: 研究什么可以被自动计算的问题;
 - **计算复杂性理论**: 研究完成一个可计算问题需要的计算时间和空间资源的代价问题
- **算法和数据结构**: 研究解决可计算问题的具体方法、步骤和过程, 及其所需要的数据表示和操作方法
- **程序设计方法学和程序设计语言**: 主要研究解决各种计算问题和软件系统设计所需要采用的程序设计原理和方法, 以及相关的程序设计语言的设计实现原理和方法

ACM/IEEE计算机分支学科

计算机科学(Computer Science, CS)

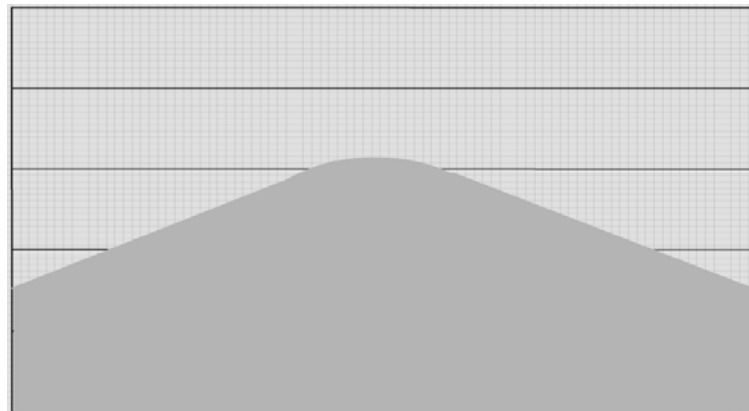
- **计算机系统的组成和结构**：主要研究计算机系统的基本硬件元素、软件元素和利用软硬件元素设计和构建计算机系统的原理和方法。
- **计算机应用基础理论和方法研究**：主要研究各种应用系统中偏向于基本理论和基础技术方法的问题，应用可包括：计算机图形学，计算机视觉，数据库，人工智能，机器人，人机交互，嵌入式系统，计算机网络通信，信息安全，普适计算等等，以及在生物、物理、化学、天文、气象、经济等各学科的数值或符号计算问题。

ACM/IEEE计算机分支学科

计算机工程 (Computer Engineering, CE)

着重研究计算机以及计算机系统的设计 and 构建，涉及软件、硬件、通信以及它们之间相互作用的研究；侧重于传统的电子工程、数学以及用它们解决计算机和计算机系统设计方面的理论、原理与工程实践。

企业组织和信息系统
应用技术
软件技术
系统平台构架
计算机硬件体系



理论
原理
创新

技术
工程
系统

应用
部署
配置

理论

技术

应用

ACM/IEEE计算机分支学科

计算机工程 (Computer Engineering, CE)

主要研究领域包括:

- **计算机体系结构技术**: 主要研究单处理器的指令集体系结构、指令级和运算部件级高级流水线技术
- **多核/多处理器并行处理技术**: 研究基于多核/多处理器的并行处理构架和并行计算技术, 包括对称多处理器 (SMP)、集群计算 (Clustering)、网格计算 (Grid) 等并行处理体系结构和计算机技术
- **嵌入式系统**: 研究开发内嵌专用处理器和软件的各种设备和系统, 如手机、数字音频播放器、数字摄像机等娱乐设备、家用电器、汽车、制造设备与装备、医疗设备、武器装备等。

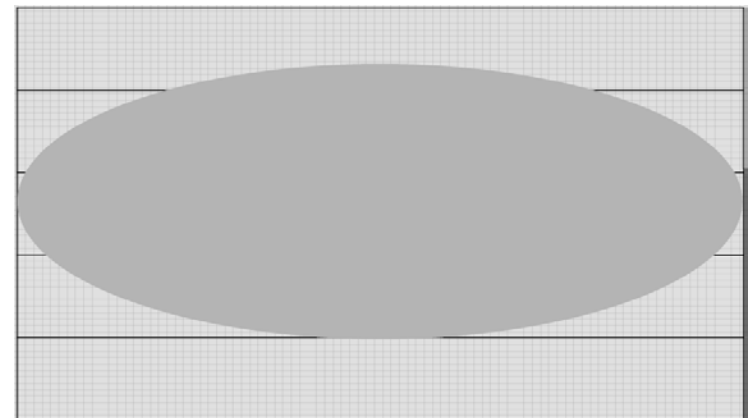
ACM/IEEE计算机分支学科

软件工程 (Software Engineering, SE)

研究开发和维护可靠、有效、满足用户需求的软件系统相关的理论、工程技术方法和工具平台。软件工程将数学和计算机科学的理论与开发特定软件系统的工程实践相结合，因而计算机科学和软件工程有很多课程相同。但软件工程

专业向学生提供具体的工程知识和经验，要求学生参加有实用意义的软件开发。

企业组织和信息系统
应用技术
软件技术
系统平台构架
计算机硬件体系



理论
原理
创新

技术
工程
系统

应用
部署
配置

理论

技术

应用

ACM/IEEE计算机分支学科

软件工程 (Software Engineering, SE)

主要研究内容包括：

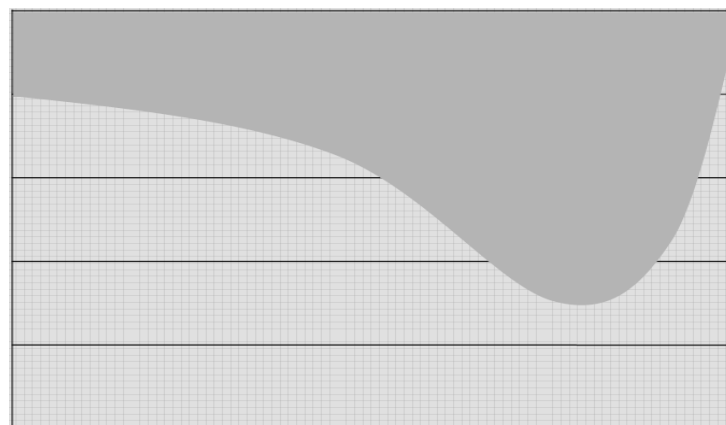
- **软件开发过程**：研究软件开发生命周期中的需求分析、软件设计、软件编码实现、软件调试与测试、软件实施与维护等基本过程，开发过程中需求和开发方各类人员的支持活动过程，以及人员、进度、质量、成本等管理活动过程
- **软件开发方法**：研究实现系统分析和设计的途径、原则、方法、技术和步骤。典型的软件开发方法有：结构化方法，面向数据结构方法、以及面向对象方法等
- **软件开发工具**：辅助计算机软件开发、运行、维护、管理、支持等过程的各种软件工具，以节省开发时间和费用，提高软件生产率和质量，如需求分析工具、概要设计工具、详细设计工具、编码工具、测试工具和维护理解工具、项目管理工具、配置管理工具、软件质量工具以及文档表格工具等。

ACM/IEEE计算机分支学科

信息系统 (Information System, IS)

当今各行业都依靠信息技术，使用自己的信息系统。信息系统关注如何将信息技术解决方案与企业和机构的业务过程相结合，通过开发和使用信息系统，帮助企业和机构完成日常的生产经营和事务管理，满足企业和机构的信息需求，有效地达到企业的目标，提高企业的生产和管理效率。

企业组织和信息系统
应用技术
软件技术
系统平台构架
计算机硬件体系



理论
原理
创新

技术
工程
系统

应用
部署
配置

理论

技术

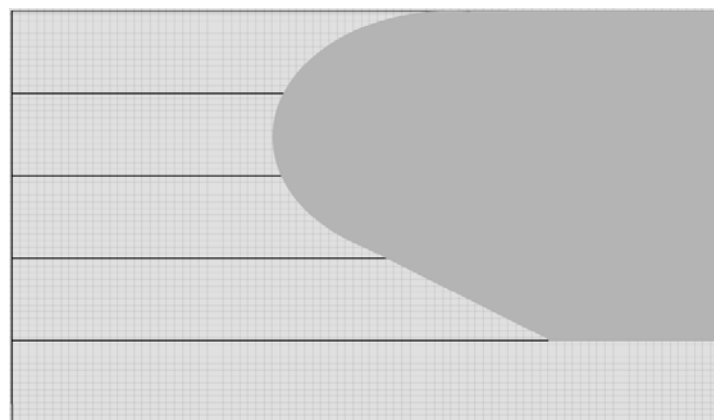
应用

ACM/IEEE计算机分支学科

信息技术 (Information Technology, IT)

这里指计算机学科下的一个专业，培养的学生能满足商业、政府、医疗、教育等各种机构对计算技术的需求。信息技术主要是研究在企业机构内规划、部署、设计、构建、安装、使用和维护计算机基础设施所需要的各种支撑和平台技术，如组建网络，管理网络及其安全，设计网页，开发多媒体资源，安装通信设备，管理电子邮件系统，以及规划和管理信息系统等。

企业组织和信息系统
应用技术
软件技术
系统平台构架
计算机硬件体系



理论
原理
创新

技术
工程
系统

应用
部署
配置

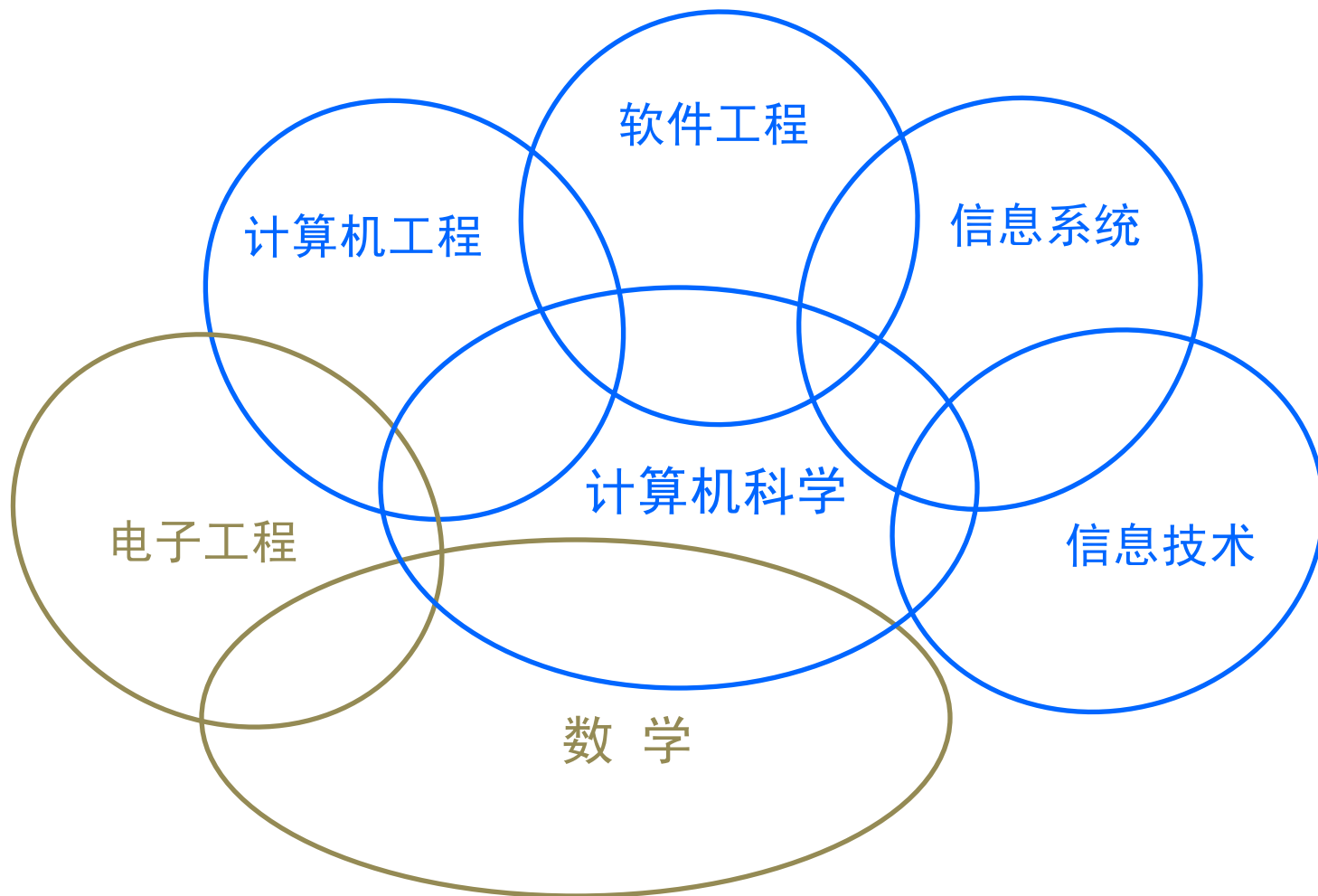
理论

技术

应用

ACM/IEEE计算机分支学科

分支学科之间的关系



计算机学科知识体系

计算机学科知识体系分为14个知识领域

- 离散结构 DS: Discrete Structures (DS)
- 程序设计基础 PF: Programming Fundamentals (PF)
- 算法和复杂性 AL: Algorithms and Complexity (AL)
- 体系结构和组织 AR: Architecture and Organization (AR)
- 操作系统 OS: Operating Systems (OS)
- 网络计算 NC: Net-Centric Computing (NC)
- 程序设计语言 PL: Programming Languages (PL)
- 人机交互 HC: Human-Computer Interaction (HC)
- 图形和可视化计算 GV: Graphics and Visual Computing (GV)
- 智能系统 IS: Intelligent Systems (IS)
- 信息管理 IM: Information Management (IM)
- 社会和职业问题 SP: Social and Professional Issues (SP)
- 软件工程 SE: Software Engineering (SE)
- 计算科学和数值计算 CN: Computational Science and Numerical Methods (CN)

计算机学科知识体系

不同专业公共的核心知识

计算机 科学	计算机 工程	软件 工程	信息 系统	信息 技术
不同专业公共的核心知识				

计算机学科知识体系

所有专业必修的核心知识(一)(下划线部分)

数字为相应核心知识需要的最小课堂学习时间
ACM/IEEE建议课外学习时间通常是核心时间的3倍

DS. Discrete Structures (43 core hours)

DS1. Functions, relations, and sets (6)

DS2. Basic logic (10)

DS3. Proof techniques (12)

DS4. Basics of counting (5)

DS5. Graphs and trees (4)

DS6. Discrete probability (6)

PF. Programming Fundamentals (38 core hours)

PF1. Fundamental programming constructs (9)

PF2. Algorithms and problem-solving (6)

PF3. Fundamental data structures (14)

PF4. Recursion (5)

PF5. Event-driven programming (4)

AL. Algorithms and Complexity (31 core hours)

AL1. Basic algorithmic analysis (4)

AL2. Algorithmic strategies (6)

AL3. Fundamental computing algorithms (12)

AL4. Distributed algorithms (3)

AL5. Basic computability (6)

AL6. The complexity classes P and NP

AL7. Automata theory

AL8. Advanced algorithmic analysis

AL9. Cryptographic algorithms

AL10. Geometric algorithms

AL11. Parallel algorithms

AR. Architecture and Organization (36 core hours)

AR1. Digital logic and digital systems (6)

AR2. Machine level representation of data (3)

AR3. Assembly level machine organization (9)

AR4. Memory system organization and architecture (5)

AR5. Interfacing and communication (3)

AR6. Functional organization (7)

AR7. Multiprocessing and alternative architectures (3)

AR8. Performance enhancements

AR9. Architecture for networks and distributed systems

OS. Operating Systems (18 core hours)

OS1. Overview of operating systems (2)

OS2. Operating system principles (2)

OS3. Concurrency (6)

OS4. Scheduling and dispatch (3)

OS5. Memory management (5)

OS6. Device management

OS7. Security and protection

OS8. File systems

OS9. Real-time and embedded systems

OS10. Fault tolerance

OS11. System performance evaluation

OS12. Scripting

计算机学科知识体系

所有专业必修的核心知识(二)(下划线部分)

NC. Net-Centric Computing (15 core hours)

- NC1. Introduction to net-centric computing (2)
- NC2. Communication and networking (7)
- NC3. Network security (3)
- NC4. The web as an example of client-server computing (3)
- NC5. Building web applications
- NC6. Network management
- NC7. Compression and decompression
- NC8. Multimedia data technologies
- NC9. Wireless and mobile computing

PL. Programming Languages (21 core hours)

- PL1. Overview of programming languages (2)
- PL2. Virtual machines (1)
- PL3. Introduction to language translation (2)
- PL4. Declarations and types (3)
- PL5. Abstraction mechanisms (3)
- PL6. Object-oriented programming (10)
- PL7. Functional programming
- PL8. Language translation systems
- PL9. Type systems
- PL10. Programming language semantics
- PL11. Programming language design

HC. Human-Computer Interaction (8 core hours)

- HC1. Foundations of human-computer interaction (6)
- HC2. Building a simple graphical user interface (2)
- HC3. Human-centered software evaluation
- HC4. Human-centered software development
- HC5. Graphical user-interface design
- HC6. Graphical user-interface programming
- HC7. HCI aspects of multimedia systems
- HC8. HCI aspects of collaboration and communication

GV. Graphics and Visual Computing (3 core hours)

- GV1. Fundamental techniques in graphics (2)
- GV2. Graphic systems (1)
- GV3. Graphic communication
- GV4. Geometric modeling
- GV5. Basic rendering
- GV6. Advanced rendering
- GV7. Advanced techniques
- GV8. Computer animation
- GV9. Visualization
- GV10. Virtual reality
- GV11. Computer vision

计算机学科知识体系

所有专业必修的核心知识(三)(下划线部分)

IS. Intelligent Systems (10 core hours)

- IS1. Fundamental issues in intelligent systems (1)
- IS2. Search and constraint satisfaction (5)
- IS3. Knowledge representation and reasoning (4)
- IS4. Advanced search
- IS5. Advanced knowledge representation and reasoning
- IS6. Agents
- IS7. Natural language processing
- IS8. Machine learning and neural networks
- IS9. AI planning systems
- IS10. Robotics

IM. Information Management (10 core hours)

- IM1. Information models and systems (3)
- IM2. Database systems (3)
- IM3. Data modeling (4)
- IM4. Relational databases
- IM5. Database query languages
- IM6. Relational database design
- IM7. Transaction processing
- IM8. Distributed databases
- IM9. Physical database design
- IM10. Data mining
- IM11. Information storage and retrieval
- IM12. Hypertext and hypermedia
- IM13. Multimedia information and systems
- IM14. Digital libraries

SP. Social and Professional Issues (16 core hours)

- SP1. History of computing (1)
- SP2. Social context of computing (3)
- SP3. Methods and tools of analysis (2)
- SP4. Professional and ethical responsibilities (3)
- SP5. Risks and liabilities of computer-based systems (2)
- SP6. Intellectual property (3)
- SP7. Privacy and civil liberties (2)
- SP8. Computer crime
- SP9. Economic issues in computing
- SP10. Philosophical frameworks

SE. Software Engineering (31 core hours)

- SE1. Software design (8)
- SE2. Using APIs (5)
- SE3. Software tools and environments (3)
- SE4. Software processes (2)
- SE5. Software requirements and specifications (4)
- SE6. Software validation (3)
- SE7. Software evolution (3)
- SE8. Software project management (3)
- SE9. Component-based computing
- SE10. Formal methods
- SE11. Software reliability
- SE12. Specialized systems development

CN. Computational Science (no core hours)

- CN1. Numerical analysis
- CN2. Operations research
- CN3. Modeling and simulation
- CN4. High-performance computing

计算机学科知识体系

计算机科学专业的知识体系

- CS-AR 体系结构与组织
- CS-AL 算法与复杂性
- CS-HC 人机交互
- CS-OS 操作系统
- CS-PF 程序设计基础
- CS-SP 社会与职业问题
- CS-SE 软件工程
- CS-DS 离散结构
- CS-NC 以网络为中心的计算
- CS-PL 程序设计语言
- CS-GV 图形学与可视化计算
- CS-IS 智能系统
- CS-IM 信息管理
- CS-CN 数值计算科学

计算机学科知识体系

计算机工程专业的知识体系

- CE-ALG 算法与复杂度
- CE-CAO 体系结构和组织
- CE-CSE 计算机系统工程
- CE-CSG 电路和信号
- CE-DBS 数据库系统
- CE-DIG 数字逻辑
- CE-DSP 数字信号处理
- CE-ELE 电子学
- CE-ESY 嵌入式系统
- CE-HCI 人机交互
- CE-NWK 计算机网络
- CE-OPS 操作系统
- CE-PRF 程序设计基础
- CE-SPR 社会和职业问题
- CE-SWE 软件工程
- CE-VLS VLSI设计与构造
- CE-DSC 离散结构
- CE-PRS 概率和统计

计算机学科知识体系

软件工程专业知识体系

- SE-CMP 计算基础
- SE-FND 数学和工程基础
- SE-PRF 职业实践
- SE-MAA 软件建模与分析
- SE-DES 软件设计
- SE-VAV 软件验证与确认
- SE-EVO 软件进化
- SE-PRO 软件过程
- SE-QUA 软件质量
- SE-MGT 软件管理

计算机学科知识体系

各知识点在不同专业中的权重和要求（一）

知识点	CS		CE		SE	
	最少	最多	最少	最多	最少	最多
程序设计基础	5	5	4	4	5	5
算法与复杂性	5	5	2	4	4	4
计算机组织结构	2	4	5	5	2	4
操作系统原理与设计	3	5	2	4	3	4
操作系统配置与使用	2	4	2	3	2	4
以网络为中心的资源配置与使用	2	3	1	2	2	3
程序设计语言理论	3	5	1	2	2	4
人机交互	2	4	2	5	3	5
图形与可视化理论	1	5	1	3	1	3
智能系统	2	5	1	3	0	0
信息管理(数据库)理论	2	5	1	3	2	5
信息管理(数据库)实践	1	4	1	2	1	4
法律/职业/伦理/社会	2	4	2	5	2	5
信息系统开发	0	2	0	2	2	4
项目管理	1	2	2	4	4	5
技术需求分析	0	1	0	1	0	2
软件模型与分析	2	3	1	3	4	5
软件设计	3	5	2	4	5	5
软件验证与测评	1	2	1	3	4	5

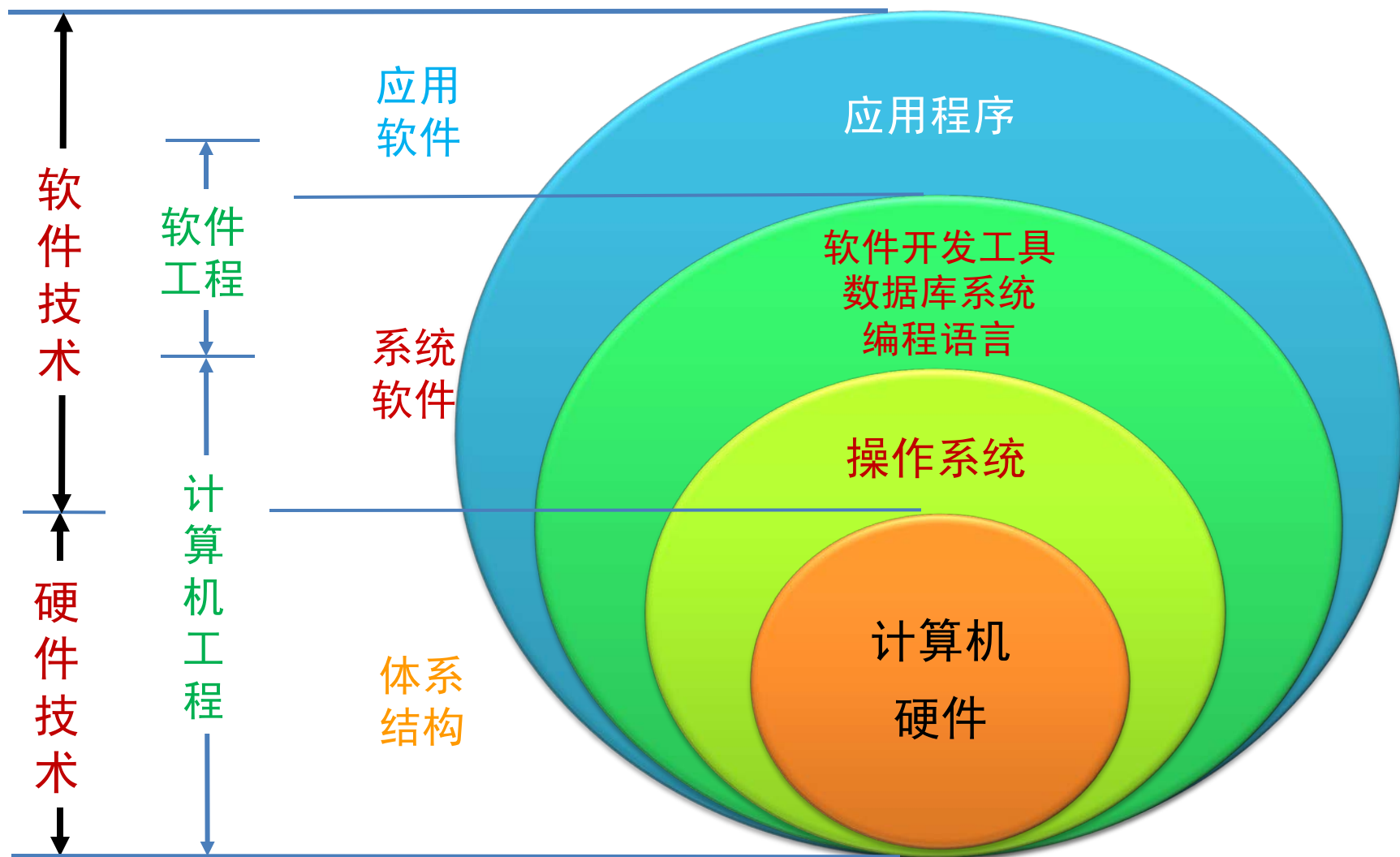
计算机学科知识体系

各知识点在不同专业中的权重和要求（二）

知识点	CS		CE		SE	
	最少	最多	最少	最多	最少	最多
计算机系统工程	1	2	5	5	2	3
嵌入式系统	1	3	2	5	2	4
电路与系统	0	2	5	5	0	0
电子工程	0	0	5	5	0	0
数字电路	2	3	5	5	1	3
分布式系统	1	3	3	5	2	4
数字信号处理	0	2	5	5	0	2
大规模集成电路设计	0	1	2	5	0	1
硬件测试与维护	0	0	3	5	0	0
信息安全:实现与管理	1	3	1	2	1	3
系统管理	1	1	1	2	1	2
系统集成与体系结构	1	2	1	4	1	3
数字媒体开发—Web系统与技术	0	1	0	2	0	1
技术支持	0	1	0	1	0	1
数学基础	4	5	5	5	4	5
人际沟通	1	3	3	4	3	4

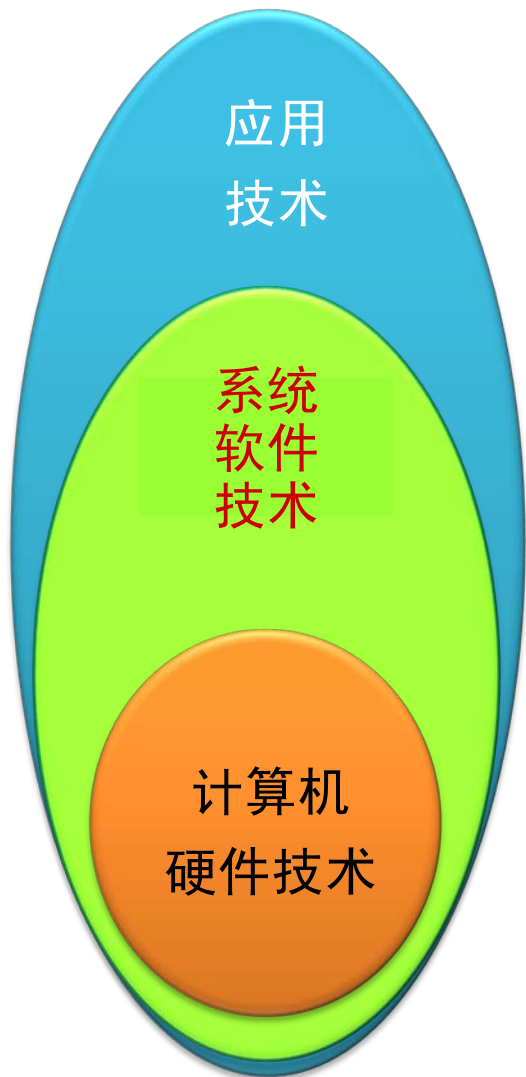
计算机学科课程体系

从计算机系统的构成看课程体系设置



计算机学科课程体系

本系主要课程设置



管理信息系统 电子商务

图形学 图像处理 多媒体

数据通信 人工智能 计算机安全

软件工程 软件体系结构 专业必选课

编译原理 高级程序设计 数据库

操作系统 数据结构 算法分析

网络技术 专业核心课 I/O接口 嵌入式系统

计算机组成与结构 计算机组成与结构实验

数字逻辑电路 数字逻辑实验 专业平台课

计算机概论 程序设计基础 离散数学

单考的考研课程

组合的专业考研课程 (150分):
组成原理45分
数据结构45分
操作系统30分
网络技术30分

本系加试课程
离散数学80分
编译原理70分

概率统计

数理逻辑

数学

大类
培养
阶段

ABC类课程

大学外语、大学体育

64
(59/5)

识
通
修

数学第一层次

大学物理 (实验)

专业
培养
阶段

计算机系统概论

专业准入 18

学
科
平
台

离散数学

程序设计基础

数字逻辑电路 (实验)

数据结构

计算机组成原理 (实验)

专业准出 24

专
业
核
心

算法设计与分析

计算机网络 (实验)

操作系统

多元
培养
阶段

高级程序设计

编译原理

数据库概论

12

必
选

概率与数理统计

多媒体技术

软件工程

嵌入式系统

指
选

数理逻辑

8

数据挖掘初步

Java程序设计

9

软件体系结构

计算方法

人工智能

19

数据通信

数字图像处理

18

面向对象设计方法

并行处理技术

计算机图形学

数学建模

计算机安全

数字信号处理

微机原理与接口

计算机程序设计语言

随机算法

管理信息系统

电子商务

LINUX系统分析

形式语言与自动机

Internet软件新技术

网络安全试验

软件测试

计算机网络编程

计算机学科课程体系

- 通修课程：
 - 全校公共课：大学数学、外语、政治、体育等
 - 50-70学分
- 专业（学科）核心课
 - 重要的专业基础和主干课程
 - 40-50学分
- 选修课程
 - 公共选修和文化素质课：书法、影视欣赏…
 - 专业选修课：大致分方向，软件与理论、应用技术、工程等
 - 50-60学分

计算机学科课程体系

学校“三三”本科培养体系

- 课程分模块

- 平台课（必修）、准出课（必修）、限选课（必修）
- 专业指选课、专业任选课

- 培养分阶段

- 通识阶段、专业阶段、多元阶段

- 出口分类型

- 科研型、创业/就业型、交叉融合型

计算机学科课程体系

学分

- 必修和选修的区别
- 不一定学习所有课程，但课程之间有“序”关系
- 课程持续一个学期甚至更短
- 合格即获得相应“学分”
 - 必修课学分不可缺少
 - 某些类型的学分必须达标
 - 四年期满，学分“足够”才可毕业
- 学分绩和平均学分绩
- 学科交叉培养预留学分空间：26-32学分
- 完成准出课程后，可以在此空间内去外系选修课程，直到完成总学分

计算机学科课程体系

毕业、结业和肄业

- 毕业：德、体合格，按时完成学籍管理规定的通修课程和学科课程，总学分数达到毕业要求
- 结业：规定学制内学分要求没有达标、体育不合格、留校察看处分未终止
- 肄业：未完成学制，终止学业

学位

- 正常毕业且学位英语过关，学士学位
- 学位证书和毕业证书，是走向市场、走向社会的基本装备！

计算机学科课程体系

常见违规及其处罚和后续影响

— 考试作弊：

- 0分计算
- 记过及以上处分，不授予学位
- 两次作弊，开除学籍

— 代考或利用通讯工具作弊：

- 开除学籍

— 旷课

- 10-30学时：警告处分； 31-49学时：记过； 超过50学时，退学

计算机学科课程体系

学分数不够

- 第一学年不足28学分
- 前两年累计不足56学分
- 前三年累计不足84学分
- 前4年累计不足112学分
- 退学试读（无学位证书）或者退学

计算机技术学习内容和层次

理论基础/研究能力

实践能力

博士

前沿理论技术
研究创新

高级理论技术

业

界

高级的系统
设计开发能力

高级编程技术

硕士

基本理论技术
研究能力

中高级理论技术

界

界

技

产

基本的系统
设计开发能力

中级编程技术

本科

基本理论/
原理/技术

术

工

具

/

平

台

基本编程技术

大中专、职业学校

基本制作技术

社会大众

操作使用技术

问题与讨论

欢迎同学们发送邮件提出问题
电子邮件:yhuang@nju.edu.cn