

2.3. Descriptions of the Major Computing Disciplines

In this section, we characterize each of the five major computing disciplines. See Sections 3.4 and 3.5 for more information on how to understand this important distinction between the names of the computing disciplines and the names of a particular degree program.

2.3.1. *Computer Engineering*

Computer engineering is concerned with the design and construction of computers and computer-based systems. It involves the study of hardware, software, communications, and the interaction among them. Its curriculum focuses on the theories, principles, and practices of traditional electrical engineering and mathematics and applies them to the problems of designing computers and computer-based devices.

Computer engineering students study the design of digital hardware systems including communications systems, computers, and devices that contain computers. They study software development, focusing on software for digital devices and their interfaces with users and other devices. CE study may emphasize hardware more than software or there may be a balanced emphasis. CE has a strong engineering flavor.

Currently, a dominant area within computing engineering is embedded systems, the development of devices that have software and hardware embedded in them. For example, devices such as cell phones, digital audio players, digital video recorders, alarm systems, x-ray machines, and laser surgical tools all require integration of hardware and embedded software and all are the result of computer engineering.

2.3.2. *Computer Science*

Computer science spans a wide range, from its theoretical and algorithmic foundations to cutting-edge developments in robotics, computer vision, intelligent systems, bioinformatics, and other exciting areas. We can think of the work of computer scientists as falling into three categories.

- They design and implement software. Computer scientists take on challenging programming jobs. They also supervise other programmers, keeping them aware of new approaches.
- They devise new ways to use computers. Progress in the CS areas of networking, database, and human-computer-interface enabled the development of the World Wide Web. Now CS researchers are working with scientists from other fields to make robots become practical and intelligent aides, to use databases to create new knowledge, and to use computers to help decipher the secrets of our DNA.
- They develop effective ways to solve computing problems. For example, computer scientists develop the best possible ways to store information in databases, send data over networks, and display complex images. Their theoretical background allows them to determine the best performance possible, and their study of algorithms helps them to develop new approaches that provide better performance.

Computer science spans the range from theory through programming. Curricula that reflect this breadth are sometimes criticized for failing to prepare graduates for specific jobs. While other disciplines may produce graduates with more immediately relevant job-related skills, computer science offers a comprehensive foundation that permits graduates to adapt to new technologies and new ideas.

2.3.3. Information Systems

Information systems specialists focus on integrating information technology solutions and business processes to meet the information needs of businesses and other enterprises, enabling them to achieve their objectives in an effective, efficient way. This discipline's perspective on information technology emphasizes information, and views technology as an instrument for generating, processing, and distributing information. Professionals in the discipline are primarily concerned with the information that computer systems can provide to aid an enterprise in defining and achieving its goals, and the processes that an enterprise can implement or improve using information technology. They must understand both technical and organizational factors, and they must be able to help an organization determine how information and technology-enabled business processes can provide a competitive advantage.

The information systems specialist plays a key role in determining the requirements for an organization's information systems and is active in their specification, design, and implementation. As a result, such professionals require a sound understanding of organizational principles and practices so that they can serve as an effective bridge between the technical and management communities within an organization, enabling them to work in harmony to ensure that the organization has the information and the systems it needs to support its operations. Information systems professionals are also involved in designing technology-based organizational communication and collaboration systems.

A majority of Information Systems (IS) programs are located in business schools. All IS degrees combine business and computing coursework. A variety of IS programs exist under various labels which often reflect the nature of the program. For example, programs in Computer Information Systems usually have the strongest technology focus, while programs in Management Information Systems emphasize the organizational and behavioral aspects of IS. Degree program names are not always consistent.

2.3.4. Information Technology

Information technology is a label that has two meanings. In the broadest sense, the term information technology is often used to refer to all of computing. In academia, it refers to undergraduate degree programs that prepare students to meet the computer technology needs of business, government, healthcare, schools, and other kinds of organizations. In some nations, other names are used for such degree programs.

In the previous section, we said that Information Systems focuses on the information aspects of information technology. Information Technology is the complement of that perspective: its emphasis is on the technology itself more than on the information it conveys. IT is a new and rapidly growing field that started as a grassroots response to the practical, everyday needs of business and other organizations. Today, organizations of every kind are dependent on information technology. They need to have appropriate systems in place. These systems must work properly, be secure, and upgraded, maintained, and replaced as appropriate. Employees throughout an organization require support from IT staff who understand computer systems and their software and are committed to solving whatever computer-related problems they might have. Graduates of information technology programs address these needs.

Degree programs in information technology arose because degree programs in the other computing disciplines were not producing an adequate supply of graduates capable of handling these very real needs. IT programs exist to produce graduates who possess the right combination of knowledge and practical, hands-on expertise to take care of both an organization's information technology infrastructure and the people who use it. IT specialists assume responsibility for selecting hardware and software products

appropriate for an organization, integrating those products with organizational needs and infrastructure, and installing, customizing, and maintaining those applications for the organization's computer users. Examples of these responsibilities include the installation of networks; network administration and security; the design of web pages; the development of multimedia resources; the installation of communication components; the oversight of email systems; and the planning and management of the technology lifecycle by which an organization's technology is maintained, upgraded, and replaced.

2.3.5. Software Engineering

Software engineering is the discipline of developing and maintaining software systems that behave reliably and efficiently, are affordable to develop and maintain, and satisfy all the requirements that customers have defined for them. This reflects its origins as outlined in Section 2.2.2. More recently, it has evolved in response to factors such as the growing impact of large and expensive software systems in a wide range of situations and the increased importance of software in safety-critical applications. Software engineering is different in character from other engineering disciplines due to both the intangible nature of software and the discontinuous nature of software operation. It seeks to integrate the principles of mathematics and computer science with the engineering practices developed for tangible, physical artifacts. Prospective students can expect to see software engineering presented in two contexts.

- Degree programs in computer science offer one or more software engineering courses as elements of the CS curriculum. Some offer a multi-course concentration in software engineering within CS.
- A number of institutions offer a software engineering degree program.

Degree programs in computer science and in software engineering have many courses in common. Software engineering students learn more about software reliability and maintenance and focus more on techniques for developing and maintaining software that is correct from its inception. While CS students are likely to have heard of the importance of such techniques, the engineering knowledge and experience provided in SE programs go beyond what CS programs can provide. The importance of this fact is so great that one of the recommendations of the SE report is that, during their program of study, students of SE should participate in the development of software to be used in earnest by others. SE students learn how to assess customer needs and develop usable software that meets those needs. Knowing how to provide genuinely useful and usable software is of paramount importance.

In the workplace, the term software engineer is a job label. There is no standard definition for this term when used in a job description. Its meaning varies widely among employers. It can be a title equivalent to computer programmer or a title for someone who manages a large, complex, and/or safety-critical software project. The layman must be mindful not confuse the discipline of software engineering with the ambiguous use of the term software engineer as used in employment advertisements and job titles.

2.4.1. Computer Engineering

The shaded portion in Figure 2.3 represents the computer engineering discipline. It is broad across the bottom because computer engineering covers the range from theory and principles to the practical application of designing and implementing products using hardware and software. It narrows towards the center as we move upwards because a computer engineer’s interests narrow as we move away from the hardware. By the time we get up to the level of software development, we see that the computer engineer’s interest has narrowed to the horizontal center because they care about software only inasmuch as they need it to develop integrated devices.

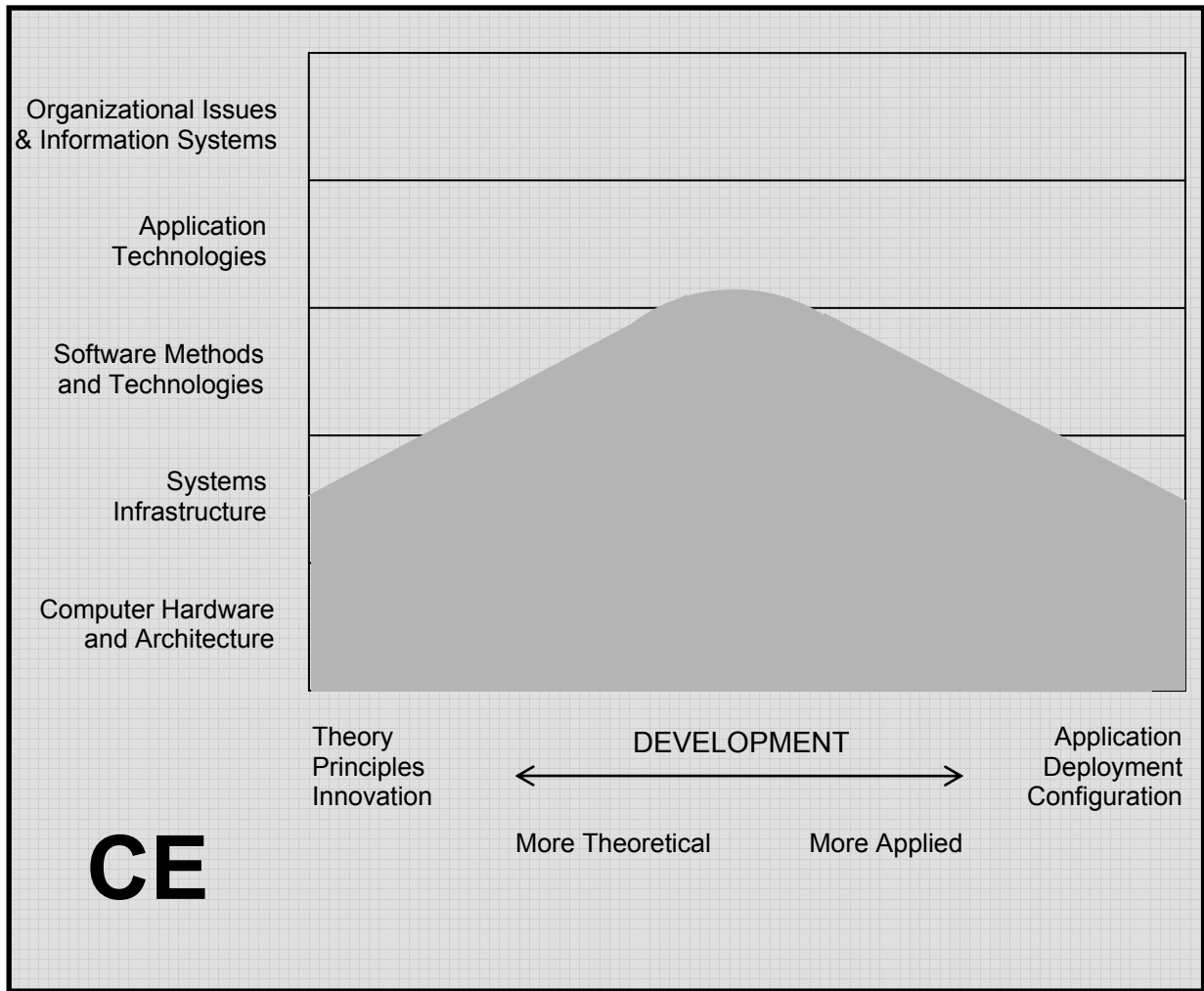


Figure 2.3. Computer Engineering

2.4.2. Computer Science

The shaded portion in Figure 2.4 represents the computer science discipline. Computer science covers most of the vertical space between the extreme top and extreme bottom because computer scientists generally do not deal with just the hardware that runs software, or just the organization that makes use of the information that computing can provide. As a group, computer scientists care about almost everything in between those areas (down as far as the software that enables devices to work and up as far as the information systems that help organizations to operate). They design and develop all types of software from systems infrastructure (operating systems, communications programs, etc.) to application technologies (web browsers, databases, search engines, etc.) Computer scientists create these capabilities, but they do not manage the deployment of them. Therefore, the shaded area for computer science narrows and then stops as we move to the right. This is because computer scientists do not help people to select computing products, or tailor products to organizational needs, or learn to use such products.

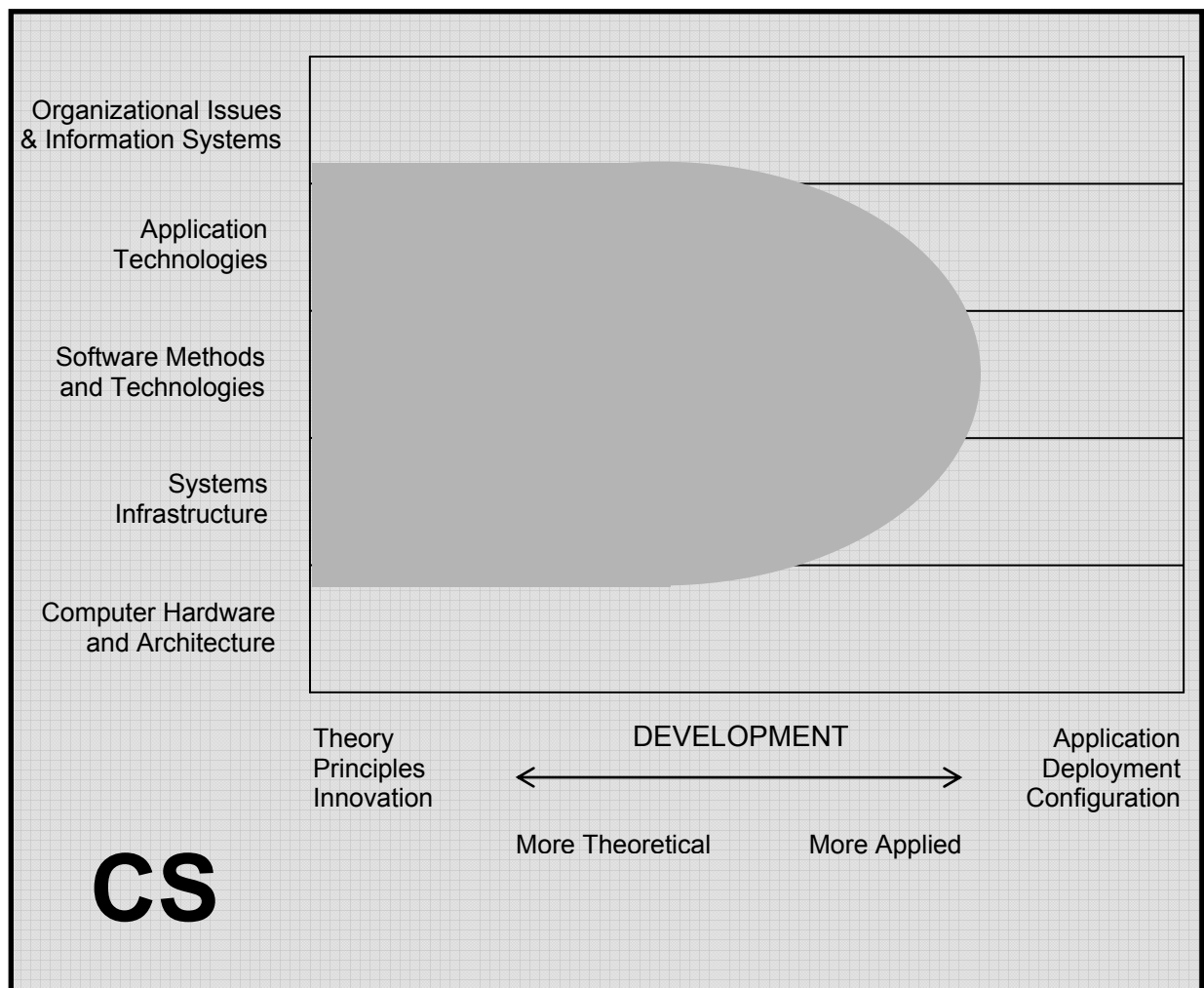


Figure 2.4. Computer Science

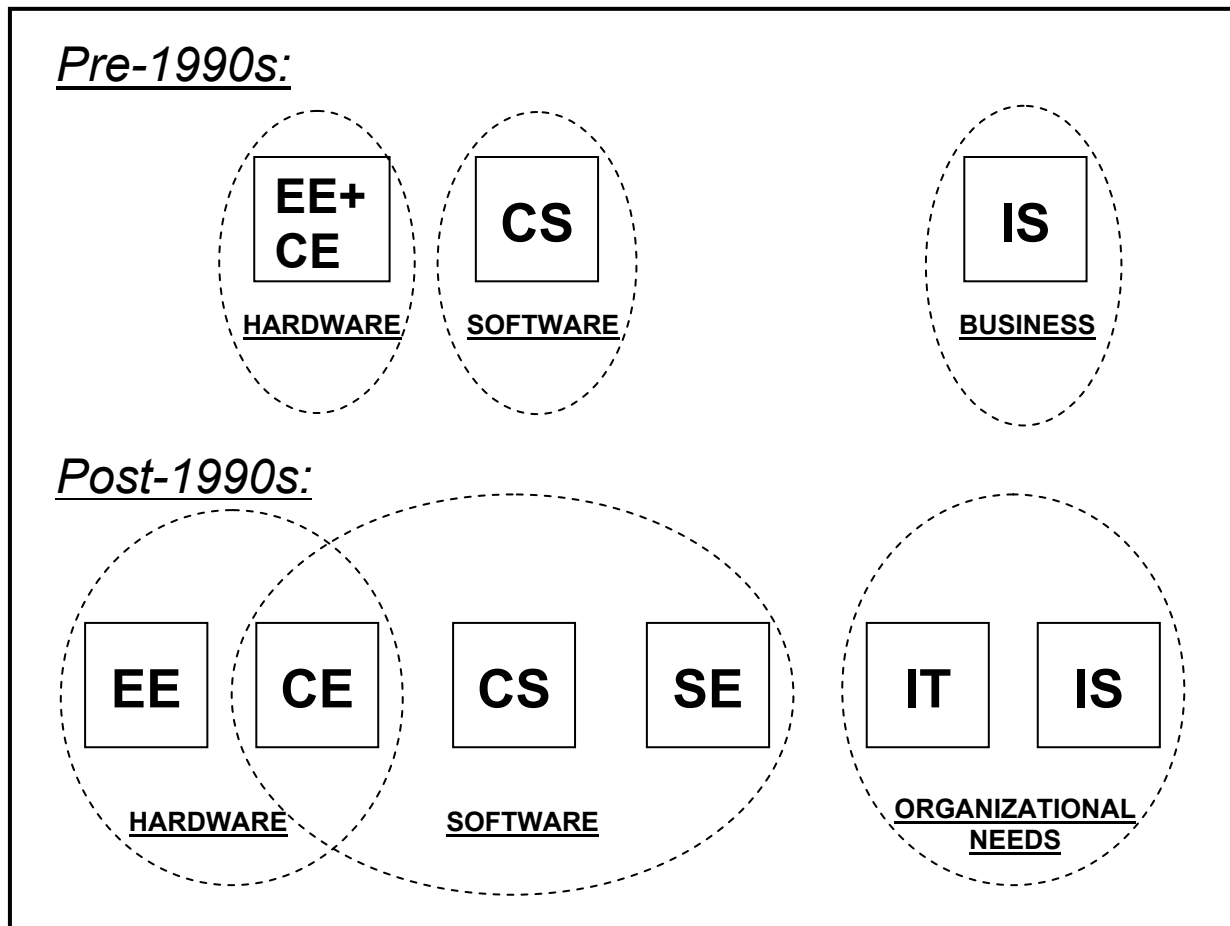


Figure 2.1. Harder Choices: How the Disciplines Might Appear to Prospective Students

Currently, there is an ongoing discussion regarding the relationship between what *computer science programs teach* and what most *graduates of computer science actually do* in their careers. To understand this discussion, it is necessary to review the characterization of computer science provided in Section 2.3.2. The work of computer scientists falls into three categories: designing and implementing software; devising new ways to use computers; and developing effective ways to solve computing problems. Let us consider what is involved in a career path in each area.

- *Career Path 1: Designing and implementing software.* This refers to the work of software development which has grown to include aspects of web development, interface design, security issues, mobile computing, and so on. This is the career path that the majority of computer science graduates choose. While a bachelor's degree is generally sufficient for entry into this kind of career, many software professionals return to school to obtain a terminal master's degree. (Rarely is a doctorate involved.) Career opportunities occur in a wide variety of settings including large or small software companies, large or small computer services companies, and large organizations of all kinds (industry, government, banking, healthcare, etc.). Degree programs in software engineering also educate students for this career path.
- *Career Path 2: Devising new ways to use computers.* This refers to innovation in the application of computer technology. A career path in this area can involve advanced graduate work, followed by a position in a research university or industrial R&D lab, or it can involve entrepreneurial activity such as was evident during the dot-com boom of the 1990s, or it can involve a combination of the two.
- *Career Path 3: Developing effective ways to solve computing problems.* This refers to the application or development of computer science theory and knowledge of algorithms to ensure the best possible solutions for computationally intensive problems. As a practical matter, a career path in the development of new computer science theory typically requires graduate work to the Ph.D. level, followed by a position in a research university or an industrial R&D laboratory.

Computer science programs generally intend to prepare students for these three career paths. In addition, there is a fourth career path that CS programs do not target but nonetheless draws many computer science graduates.

- *Career Path 4: Planning and managing organizational technology infrastructure.* This refers to the work for which the new information technology (IT) programs explicitly aim to educate students.

Of these four career paths, career paths 2 and 3 are important elements of the identity of computer science and are the kind of career paths that many computer science faculty wish to see their students choose. As a practical observation, however, only an extremely small minority of the students who earn computer science bachelor's degrees actually choose them. For those few who do, institutional lag is not an issue: a strong bachelor's degree program in computer science, followed by graduate study (probably to the doctoral level) is clearly the preferred choice.

Career paths 1 and 4 are the focus of debate. These careers draw the overwhelming majority of computer science graduates. They are also the focus of the new computing degree programs (software engineering and information technology, respectively) which have emerged to provide more focused alternatives to

Table 3.1: Comparative weight of computing topics across the five kinds of degree programs

<i>Knowledge Area</i>	<i>CE</i>		<i>CS</i>		<i>IS</i>		<i>IT</i>		<i>SE</i>	
	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>
Programming Fundamentals	4	4	4	5	2	4	2	4	5	5
Integrative Programming	0	2	1	3	2	4	3	5	1	3
Algorithms and Complexity	2	4	4	5	1	2	1	2	3	4
Computer Architecture and Organization	5	5	2	4	1	2	1	2	2	4
Operating Systems Principles & Design	2	5	3	5	1	1	1	2	3	4
Operating Systems Configuration & Use	2	3	2	4	2	3	3	5	2	4
Net Centric Principles and Design	1	3	2	4	1	3	3	4	2	4
Net Centric Use and configuration	1	2	2	3	2	4	4	5	2	3
Platform technologies	0	1	0	2	1	3	2	4	0	3
Theory of Programming Languages	1	2	3	5	0	1	0	1	2	4
Human-Computer Interaction	2	5	2	4	2	5	4	5	3	5
Graphics and Visualization	1	3	1	5	1	1	0	1	1	3
Intelligent Systems (AI)	1	3	2	5	1	1	0	0	0	0
Information Management (DB) Theory	1	3	2	5	1	3	1	1	2	5
Information Management (DB) Practice	1	2	1	4	4	5	3	4	1	4
Scientific computing (Numerical mthds)	0	2	0	5	0	0	0	0	0	0
Legal / Professional / Ethics / Society	2	5	2	4	2	5	2	4	2	5
Information Systems Development	0	2	0	2	5	5	1	3	2	4
Analysis of Business Requirements	0	1	0	1	5	5	1	2	1	3
E-business	0	0	0	0	4	5	1	2	0	3
Analysis of Technical Requirements	2	5	2	4	2	4	3	5	3	5
Engineering Foundations for SW	1	2	1	2	1	1	0	0	2	5
Engineering Economics for SW	1	3	0	1	1	2	0	1	2	3
Software Modeling and Analysis	1	3	2	3	3	3	1	3	4	5
Software Design	2	4	3	5	1	3	1	2	5	5
Software Verification and Validation	1	3	1	2	1	2	1	2	4	5
Software Evolution (maintenance)	1	3	1	1	1	2	1	2	2	4
Software Process	1	1	1	2	1	2	1	1	2	5
Software Quality	1	2	1	2	1	2	1	2	2	4
Comp Systems Engineering	5	5	1	2	0	0	0	0	2	3
Digital logic	5	5	2	3	1	1	1	1	0	3
Embedded Systems	2	5	0	3	0	0	0	1	0	4
Distributed Systems	3	5	1	3	2	4	1	3	2	4
Security: issues and principles	2	3	1	4	2	3	1	3	1	3
Security: implementation and mgt	1	2	1	3	1	3	3	5	1	3
Systems administration	1	2	1	1	1	3	3	5	1	2
Management of Info Systems Org.	0	0	0	0	3	5	0	0	0	0
Systems integration	1	4	1	2	1	4	4	5	1	4
Digital media development	0	2	0	1	1	2	3	5	0	1
Technical support	0	1	0	1	1	3	5	5	0	1

Again, min represents the minimum called for by the curriculum guidelines, and max represents the greatest emphasis one might expect in the typical case of a student who chooses to undertake optional work in that area or who graduates from an institution that requires its students to achieve mastery beyond that required by the curriculum reports. Because the difference between the min and max values can be large, programs with the same degree name may differ substantially because of the local choices made in determining their requirements. Both min and max values refer to what can be reasonably expected in the general case. For any individual student or degree program, the min value might be as low as zero and the max value might be as high as five, regardless of prevailing curricular standards.

3.1.1. How the Table Values Were Determined

Tables 3.1 and 3.2 represent the consensus of judgment reached by the CC2005 Joint Task Force. The task force formulated this consensus from an examination of the discipline-specific body of knowledge found in the most recent curriculum volume for each of the computing disciplines: computer engineering,

Table 3.2: Comparative weight of non-computing topics across the five kinds of degree programs

<i>Knowledge Area</i>	<i>CE</i>		<i>CS</i>		<i>IS</i>		<i>IT</i>		<i>SE</i>	
	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>	<i>min</i>	<i>max</i>
Organizational Theory	0	0	0	0	1	4	1	2	0	0
Decision Theory	0	0	0	0	3	3	0	1	0	0
Organizational Behavior	0	0	0	0	3	5	1	2	0	0
Organizational Change Management	0	0	0	0	2	2	1	2	0	0
General Systems Theory	0	0	0	0	2	2	1	2	0	0
Risk Management (Project, safety risk)	2	4	1	1	2	3	1	4	2	4
Project Management	2	4	1	2	3	5	2	3	4	5
Business Models	0	0	0	0	4	5	0	0	0	0
Functional Business Areas	0	0	0	0	4	5	0	0	0	0
Evaluation of Business Performance	0	0	0	0	4	5	0	0	0	0
Circuits and Systems	5	5	0	2	0	0	0	1	0	0
Electronics	5	5	0	0	0	0	0	1	0	0
Digital Signal Processing	3	5	0	2	0	0	0	0	0	2
VLSI design	2	5	0	1	0	0	0	0	0	1
HW testing and fault tolerance	3	5	0	0	0	0	0	2	0	0
Mathematical foundations	4	5	4	5	2	4	2	4	3	5
Interpersonal communication	3	4	1	4	3	5	3	4	3	4

computer science, information systems, information technology, and software engineering. It used the results of this examination to define the topical elements of the two tables. That examination also heavily influenced the numerical values assigned to each topic for each discipline. The discipline-specific bodies of knowledge provide some quantifiable data concerning the minimum coverage called for by each discipline for each topic. However, they do not provide information that is sufficient to permit any useful calculation of the relative emphasis each discipline places on a given topic.

Table 3.3. Relative performance capabilities of computing graduates by discipline

Area	Performance Capability	CE	CS	IS	IT	SE
Algorithms	Prove theoretical results	3	5	1	0	3
	Develop solutions to programming problems	3	5	1	1	3
	Develop proof-of-concept programs	3	5	3	1	3
	Determine if faster solutions possible	3	5	1	1	3
Application programs	Design a word processor program	3	4	1	0	4
	Use word processor features well	3	3	5	5	3
	Train and support word processor users	2	2	4	5	2
	Design a spreadsheet program (e.g., Excel)	3	4	1	0	4
	Use spreadsheet features well	2	2	5	5	3
	Train and support spreadsheet users	2	2	4	5	2
Computer programming	Do small-scale programming	5	5	3	3	5
	Do large-scale programming	3	4	2	2	5
	Do systems programming	4	4	1	1	4
	Develop new software systems	3	4	3	1	5
	Create safety-critical systems	4	3	0	0	5
	Manage safety-critical projects	3	2	0	0	5
Hardware and devices	Design embedded systems	5	1	0	0	1
	Implement embedded systems	5	2	1	1	3
	Design computer peripherals	5	1	0	0	1
	Design complex sensor systems	5	1	0	0	1
	Design a chip	5	1	0	0	1
	Program a chip	5	1	0	0	1
	Design a computer	5	1	0	0	1
Human-computer interface	Create a software user interface	3	4	4	5	4
	Produce graphics or game software	2	5	0	0	5
	Design a human-friendly device	4	2	0	1	3
Information systems	Define information system requirements	2	2	5	3	4
	Design information systems	2	3	5	3	3
	Implement information systems	3	3	4	3	5
	Train users to use information systems	1	1	4	5	1
	Maintain and modify information systems	3	3	5	4	3
Information management (Database)	Design a database mgt system (e.g., Oracle)	2	5	1	0	4
	Model and design a database	2	2	5	5	2
	Implement information retrieval software	1	5	3	3	4
	Select database products	1	3	5	5	3
	Configure database products	1	2	5	5	2
	Manage databases	1	2	5	5	2
	Train and support database users	2	2	5	5	2
IT resource planning	Develop corporate information plan	0	0	5	3	0
	Develop computer resource plan	2	2	5	5	2
	Schedule/budget resource upgrades	2	2	5	5	2
	Install/upgrade computers	4	3	3	5	3
	Install/upgrade computer software	3	3	3	5	3
Intelligent systems	Design auto-reasoning systems	2	4	0	0	2
	Implement intelligent systems	2	4	0	0	4
Networking and communications	Design network configuration	3	3	3	4	2
	Select network components	2	2	4	5	2
	Install computer network	2	1	3	5	2
	Manage computer networks	3	3	3	5	3
	Implement communication software	5	4	1	1	4
	Manage communication resources	1	0	3	5	0
	Implement mobile computing system	5	3	0	1	3
	Manage mobile computing resources	3	2	2	4	2
Systems Development Through Integration	Manage an organization's web presence	2	2	4	5	2
	Configure & integrate e-commerce software	2	3	4	5	4
	Develop multimedia solutions	2	3	4	5	3
	Configure & integrate e-learning systems	1	2	5	5	3
	Develop business solutions	1	2	5	3	2
	Evaluate new forms of search engine	2	4	4	4	4