

Information Technology 2008

Curriculum Guidelines for Undergraduate Degree Programs in Information Technology

**Association for Computing Machinery (ACM)
IEEE Computer Society**

**Barry M. Lunt (Chair)
Joseph J. Ekstrom
Sandra Gorka
Gregory Hislop
Reza Kamali
Eydie Lawson
Richard LeBlanc
Jacob Miller
Han Reichgelt**

Version: Posting Nov 2008

Table of Contents

Executive Summary	4
Acknowledgments	6
Chapter 1 Introduction	8
1.1 Overall structure of <i>Computing Curricula 2005: The Overview Report</i>	8
1.2 Overview of the process for developing this IT volume.....	8
1.3 Definition of Information Technology as an academic discipline	8
1.4 Broad goals of an IT program.....	9
1.5 Purpose and structure of this document	9
Chapter 2 The Context of This Report	11
2.1 Historical background.....	11
2.2 Evaluation of curriculum efforts	12
Chapter 3 The Information Technology Discipline	13
3.1 The emergence of Information Technology as a discipline.....	13
3.1.1 Technical changes.....	14
3.1.2 Pedagogical changes	14
3.2 The role of Information Technology within the computing disciplines	15
3.3 Characteristics of an IT graduate	16
3.4 Research in IT	18
Chapter 4 Principles	21
Chapter 5 Overview of the IT Body of Knowledge	23
5.1 Structure of the body of knowledge.....	23
5.1.1 Core and advanced outcomes.....	23
5.1.2 Assessing the time required to cover a unit	24
5.1.3 Packaging units into courses	25
5.2 Summary of the IT body of knowledge	25
Chapter 6 Overview of the Curricular Models	27
6.1 Overall structure of the model curricula	27
6.2 The role of experiential learning.....	27
6.3 Overview of the implementation strategies	28
6.3.1 Integration-first approach.....	28
6.3.2 Pillars-first approach.....	29
Chapter 7 The Core in the Curriculum	30
7.1 Overall philosophy.....	30
7.2 Expectations of the introductory curriculum	31
7.2.1 IT fundamentals	31
7.2.2 Programming fundamentals.....	32
7.2.3 Pervasive themes.....	32
7.3 Presenting the core using the integration-first approach.....	32
7.4 Presenting the core using the pillars-first approach.....	33
7.5 The core at the advanced level.....	33
7.5.1 Information Assurance and Security (IAS).....	33
7.5.2 Social and professional issues of IT.....	34
7.5.3 The integrative capstone experience	34
7.5.4 Other advanced materials.....	35

7.5.4.1 Optional IT classes.....	35
7.5.4.2 Application domain classes.....	35
7.5.4.3 Professional communications class.....	35
7.5.5 IT pervasive themes	35
7.6 Practicum in the IT curriculum	35
Chapter 8 Completing the Curriculum	37
8.1 General requirements	37
8.1.1 The scientific method.....	38
8.1.2 Familiarity with application domains	38
8.1.3 Communication skills	38
8.1.4 Working in teams.....	38
8.1.5 Becoming a contributing member of society	39
8.1.6 Pervasive themes in the IT curriculum	39
8.2 Advanced courses	39
Chapter 9 Professional Practice	41
9.1 Rationale	41
9.2 Current practice in education	42
9.3 Supporting professional practice.....	43
9.3.1 The private and public sectors	43
9.3.2 Modeling local and international work environments	44
9.3.3 Administration, faculty, and student roles	45
9.4 Incorporating professional practice into the curriculum.....	46
9.5 Assessing professional practice work	46
9.6 Certifications.....	47
Chapter 10 Characteristics of IT Graduates	48
Chapter 11 Computing Across the Curriculum	51
11.1 Process questions	52
11.1.1 Course specification.....	52
11.1.2 Course design.....	53
11.1.3 Course implementation	53
11.1.4 Course assessment	54
11.2 Course models.....	54
Chapter 12 Institutional Challenges	56
12.1 The need for local adaptation.....	56
12.2 Principles for curriculum design.....	57
12.3 The need for adequate computing resources.....	58
12.4 Attracting and retaining faculty	58
12.5 Faculty commitment to the degree program	59
12.6 Conclusion	59
Bibliography	60
Index	62
Appendix A The IT Body of Knowledge	64
Information Technology Fundamentals.....	68
Human Computer Interaction	70
Information Assurance and Security.....	75
Information Management.....	84
Integrative Programming and Technologies.....	90
Math and Statistics for IT	94

Networking	97
Programming Fundamentals	102
Platform Technologies	105
System Administration and Maintenance	109
System Integration and Architecture	112
Social and Professional Issues	117
Web Systems and Technologies	122
Appendix B IT Course Descriptions	127
Appendix C Learning Outcomes/Knowledge Areas Matrix	136

Computing Curricula Information Technology Volume

Executive Summary

This document represents the final report of the Joint Task Force on Computing Curricula – an undertaking of SIGITE (Special Interest Group on Information Technology Education) of the ACM (Association for Computing Machinery), the ACM, and the IEEE Computer Society – for four-year programs in Information Technology. This report dates back to December 2001, as described in [Chapter 2](#).

This volume of the report outlines a set of recommendations for four-year programs in Information Technology. As described in [Chapter 1](#), the full Computing Curriculum 2005 report (CC2005) consists of several volumes, each containing separate recommendations for computing disciplines, including Computer Engineering, Computer Science, Information Technology, Information Systems, and Software Engineering. Each of these separate volumes of CC2004 is under the control of separate committees and has been or is being published as it is completed.

Highlights of this report include the following:

- *The IT body of knowledge.* We have identified a body of knowledge appropriate to four-year Information Technology programs. Drawing on the structure of earlier curriculum reports (most notably the CS2001 volume), we have arranged that body of knowledge hierarchically, subdividing the field into knowledge areas, which are then broken down further into units, which are defined in terms of individual topics and learning outcomes. An overview of the body of knowledge appears in [Chapter 5](#).
- *Learning outcomes.* For each unit in the body of knowledge, we have developed a set of learning outcomes. These learning outcomes appear as part of the detailed description of the body of knowledge in [Appendix A](#). For most units, the learning outcomes are divided into core outcomes and advanced outcomes. In addition to the individual learning outcomes, the report outlines a set of characteristics that all Information Technology graduates should possess in [Chapter 10](#).
- *The IT core.* From the 85 units in the body of knowledge, we have selected 81 that contain the core material, accounting for approximately 314 hours of instruction. As noted in our statement of principles in [Chapter 4](#), we defined the core as the set of learning outcomes for which there is a broad consensus that the material is essential to a four-year degree in Information Technology. The philosophy behind the definition of the core is described in more detail in [Chapter 5](#).
- *The IT Advanced Outcomes.* The core is not a complete curriculum, and must be supplemented by additional material. This document proposes IT advanced learning outcomes that may be used to complete a curriculum. These advanced learning outcomes are usually part of units that also contain core learning outcomes, although there are some units that only have advanced outcomes defined.
- *Curriculum models.* The report identifies two approaches to undergraduate instruction in Information Technology, as described in [Chapter 6](#). Building on that foundation, [Chapter 7](#) describes the core material of the Information Technology curriculum, and [Chapter 8](#) describes the additional material necessary to constitute a complete four-year curriculum in Information Technology.
- *Course descriptions.* Appendix B contains a detailed model of two curricular approaches, including course descriptions for 33 courses.

The process of developing this report has been highly inclusive. More than 30 people have been directly involved in the focus groups established to contribute to the process. In addition, the report has been widely reviewed by academics and practitioners through a series of three public drafts. Since 2003, this volume has been continuously available at the SIGITE website (<http://www.sigite.org/>) and at the ACM curricula recommendations website (<http://www.acm.org/education/curricula.html>).

We have held plenary feedback sessions at each of the last six SIGITE conferences. This material has been presented at national and international conferences, including the American Society for Engineering Education (ASEE) in 2003 and 2004, the International Conference on Engineering and Computer Education (ICECE) in 2003, the International Conference on Engineering Education and Research (iCEER) in 2004, and the World Engineer's Convention (WEC) in 2004. Additionally, this material has also been published in the Journal of Information Technology Education (JITE) in 2004. These meetings have provided us with critically important feedback, which we have used to shape the final report.

A steering committee was formed late in 2007 (see next page and page 9), which included a representative of the ACM and the IEEE-Computer Society in the final modifications of the document.

Computing Curricula Information Technology Volume

Acknowledgements

Members of the IT Curriculum Committee - Writing Subcommittee: (in alphabetical order)

Joseph Ekstrom	Brigham Young University	Provo, UT
Sandra Gorka	Pennsylvania College of Technology	Williamsport, PA
Reza Kamali	Cameron University	Lawton, OK
Eydie Lawson	Rochester Institute of Technology	Rochester, NY
Barry Lunt	Brigham Young University	Provo, UT
Jacob Miller	Pennsylvania College of Technology	Williamsport, PA
Han Reichgelt	Southern Polytechnic State University	Marietta, GA

Members of the IT 4-Year Curriculum Committee: (in alphabetical order)

All seven of the above members of the Writing Subcommittee, plus:

Al Biles	Rochester Institute of Technology	Rochester, NY
Kitty Daniels	Pace University	New York City, NY
David Eggert	New Haven University	West Haven, CT
Gordon Goodman	Rochester Institute of Technology	Rochester, NY
Kent Jackson	Brigham Young University – Idaho	Rexburg, ID
Vladan Jovanovic	Georgia Southern University	Statesboro, GA
Keith Morneau	Capella University	Minneapolis, MN
Jim Leone	Rochester Institute of Technology	Rochester, NY
Soleda Leung	University of Cincinnati	Cincinnati, OH
Dick Malone	Macon State University	Macon, GA
Anne Mannette-Wright	St. John Fisher College	Rochester, NY
John Mendonca	Purdue University	West Lafayette, IN
Evelyn Rozanski	Rochester Institute of Technology	Rochester, NY
Rebecca Rutherford	Southern Polytechnic State University	Marietta, GA
Russell Shackelford	Association for Computing Machinery (ACM)	
Cheryl Willis	University of Houston	Houston, TX
Bill Wolfe	Calif. State University – Channel Islands	Camarillo, CA

Members of the Steering Committee: (in alphabetical order)

Joseph Ekstrom	Brigham Young University	Provo, UT
Greg Hislop	Drexel University	Philadelphia, PA
Richard LeBlanc	Georgia Tech	Atlanta, GA
Barry Lunt	Brigham Young University	Provo, UT

Editor and Chair of above committees:

Barry Lunt	Brigham Young University	Provo, UT
------------	--------------------------	-----------

Acknowledgement of Support and Other Participation

The IT 4-Year Curriculum Committee and the IT 4-Year Curriculum Writing Subcommittee would like to thank the ACM and their representative Russell Shackelford for generous support. Their providing of a substantial grant made much of this work possible. Additionally, the participation of members of the ACM Education Board and the IEEE Computer Society made it possible to identify and complete final changes.



Computing Curricula Information Technology Volume

Chapter 1 Introduction

In the fall of 2003, the Special-Interest Group for Information Technology Education (SIGITE) of the Association for Computing Machinery (ACM) established an IT Curriculum Writing Subcommittee to undertake the task of drafting a preliminary version of an IT volume for inclusion within the curriculum structure for the computing disciplines included in *Computing Curricula 2005: The Overview Report*. The charter of this committee was to take the material already created by the SIGITE Curriculum Committee, augment it as necessary, and organize it into a form acceptable as the Information Technology volume of the *Computing Curricula Series*.

1.1 Overall structure of *Computing Curricula 2005: The Overview Report*

In light of the broadening scope of computing, the Joint Task Force for Computing Curricula 2005 (a cooperative project of the ACM, AIS, and IEEE-CS) was appointed to produce a volume describing five computing disciplines and their relationship to each other. This volume would pull together the curricular recommendations for Computer Engineering, Computer Science, Information Systems, Information Technology (this volume), and Software Engineering. A draft of *CC 2005: The Overview Report* was used in the early development of this volume; the final version of *CC 2005: The Overview Report* has also been used in completing this volume.

1.2 Overview of the process for developing this IT volume

We, the authors of this document, acknowledge the process used in developing the structure and content of the CC2001 document, and have made every effort to have the same structure in this document. Additionally, some material has been incorporated from *CC 2005: The Overview Report*.

Developing the recommendations in this volume has been primarily the responsibility of the SIGITE Curriculum Committee-Writing Subcommittee and the Steering Committee, the members of which are listed at the beginning of this report.

1.3 Definition of Information Technology as an academic discipline

Information Technology (IT) in its broadest sense encompasses all aspects of computing technology. IT, as an academic discipline, is concerned with issues related to **advocating for users** and meeting their needs within an organizational and societal context through the **selection, creation, application, integration and administration** of computing technologies. Chapters 3 and 5 expand this definition.

1.4 Broad goals of an IT program

IT programs aim to provide IT graduates with the skills and knowledge to take on appropriate professional positions in Information Technology upon graduation and grow into leadership positions or pursue research or graduate studies in the field. Specifically, within five years of graduation a student should be able to:

1. Explain and apply appropriate information technologies and employ appropriate methodologies to help an individual or organization achieve its goals and objectives;
2. Function as a user advocate;
3. Manage the information technology resources of an individual or organization;
4. Anticipate the changing direction of information technology and evaluate and communicate the likely utility of new technologies to an individual or organization;
5. Understand and, in some cases, contribute to the scientific, mathematical and theoretical foundations on which information technologies are built;
6. Live and work as a contributing, well-rounded member of society.

In item #2 above, it should be recognized that in many situations, "a user" is not a homogeneous entity. Students should recognize that the role of user advocate is often complicated by the fact that different users have different and sometimes contradictory interests and goals. For example, among the people who might be included in the category of user and who often have different goals are: 1) clerks who are hands-on users of computers; 2) professionals and analysts who are hands-on users of computers; 3) people who are users of information generated through computers; 4) managers and executives who have views about how computers and computer applications should be deployed and used in their organizations; 5) internal and external customers of IT-reliant work systems; and 6) other stakeholders who care about the situation for a variety of reasons

1.5 Purpose and structure of this document

The primary purpose of this document is to set out a model curriculum that enables students to acquire the skills necessary to achieve the goals in Section 1.4. It is intended as a guide for institutions of higher education in the creation and/or revision of four-year programs in IT. It is expected that it will also be useful in the creation and/or revision of two-year programs in IT.

This document is intended to describe a typical curriculum for an IT program. It is fully anticipated that there will be many variations and flavors of IT programs, with more emphasis in one or several knowledge areas, and less in others.

It is important to understand how this variation is represented in the description of the body of knowledge. Because of the broad, integrative nature of IT, the core includes basic coverage of a high percentage of all the units in the body of knowledge (83 out of 88 units). The implication is that much of the learning beyond the core comes not from other units, but from students gaining additional depth in the same units. In order to reflect this structure, units are defined with both core learning outcomes and advanced learning outcomes.

The expectation is that every IT degree will include the entire set of core learning outcomes plus a substantial, but varying, set of advanced learning outcomes. This contrasts with other computing disciplines where the core consists of a smaller proportion of the full set of units. For example, CC2001 (for Computer Science) defines a core of 64 out of 128 topics, and there is only one set of learning outcomes per unit.

The main body of this report consists of 12 chapters. [Chapter 2](#) begins with a brief history of the development of this report and the context within which the development took place. [Chapter 3](#) outlines the changes that have recently occurred in computing which gave rise to the IT discipline, and discusses the implications that those changes have for curriculum design and pedagogy. In [Chapter 4](#), we articulate a set of principles that have guided the development of this volume of the Computing Curricula report, as we have attempted to build on the strengths of our predecessors while avoiding some of the problems observed in earlier reports or volumes. [Chapters 5](#) and [6](#) present overviews of the Information Technology body of knowledge and the curriculum recommendations that are presented in detail in the appendices. [Chapter 7](#) describes the core courses and approaches we recommend for an IT program. Because these courses alone do not constitute a complete four-year curriculum, [Chapter 8](#) summarizes additional courses and topics that must be included as part of the academic program. One important aspect of the complete curriculum involves the study of professional practice, which is discussed in [Chapter 9](#). In [Chapter 10](#), we outline a set of characteristics that define the successful Information Technology graduate. [Chapter 11](#) looks at the problem of teaching Information Technology and computing-related skills to students in other disciplines. Finally, [Chapter 12](#) offers strategic and tactical suggestions for addressing the institutional challenges that affect the implementation of this report.

The bulk of the material in this report appears in two appendices. [Appendix A](#) looks in detail at the body of knowledge for four-year Information Technology programs. [Appendix B](#) consists of full descriptions for the recommended courses that comprise the sample curricula. [Appendix C](#) lists each of the knowledge areas and units that address each of the learning outcomes required for accreditation. We hope that providing both the body of knowledge and course descriptions helps institutions to create effective curricula more easily than using either of these sources alone.



Computing Curricula Information Technology Volume

Chapter 2 The Context of This Report

In developing this report, the IT Curriculum Committee did not have to start from scratch. We have benefited tremendously from past curriculum studies, and are indebted to the authors of those studies and to the authors of the CS volume of the *CC2001* report and *CC 2005: The Overview Report* for their dedicated efforts. As part of our early work on this *Computing Curricula – IT Volume*, we looked carefully at the most recent curriculum studies – particularly *Computing Curricula 2001* and *CC 2005: The Overview Report* – to get a sense of how those studies have influenced computing education. By identifying which aspects of the previous reports have been the most successful, we hoped to structure this volume to maximize its impact. This chapter provides a historical overview and the context within which this report has been developed.

2.1 Historical background

The past four decades have seen the computing field expand dramatically, from a small group of academics mostly in mathematics and electrical engineering, to a full academic discipline known as computer science, to even more computer-related disciplines in the last decade. The computer-related disciplines defined in the five volumes of the Computing Curricula report include Computer Science, Computer Engineering, Information Systems, Information Technology, and Software Engineering. We anticipate that there may be others in the future.

Efforts to formally define the IT curriculum began in the Fall of 2001 with informal meetings between faculty in IT programs at a small number of institutions. This effort brought forth its first fruit in the first Conference on Information Technology Curriculum (CITC-1), hosted by Brigham Young University in Provo, UT in December 2001. This conference included 34 representatives from 15 institutions of higher education with programs in Information Technology, as well as 7 representatives from 4 professional and accreditation organizations. The main topics of this conference were defining the topics in an IT curriculum and preparing for accreditation.

The major outcomes of this conference included: 1) a list of approximately 30 topics, with their related subtopics, that should be part of a 4-year education in Information Technology; 2) a committee to work on curriculum issues; 3) a committee to work on accreditation issues; and 4) a parent organization (the Society for Information Technology Education, or SITE) for all people interested.

Between 2001 and 2003, three more CITC conferences were held, and both the accreditation and curriculum committees worked on their assigned tasks. Many educational institutions sought input from members of their industrial advisory boards. In July of 2003, SITE became SIGITE, the Special Interest Group for Information Technology Education, of the ACM. Draft criteria for accreditation were approved by SITE and were posted for public approval in the Fall of 2003. After approval of the general computing criteria, three IT programs were successfully accredited as general computing programs in 2005. The IT-specific criteria were approved in 2005 and

in 2006, one IT program was accredited specifically as an IT program. Since then, other IT programs have also been accredited under the IT-specific criteria.

The curriculum committee began working on this document in December of 2003, using as a model the CS volume of the *CC2001* document. A draft of *CC 2005: The Overview Report* became available in 2005, and this was used also. In October 2005, a draft of this IT Volume was posted on the ACM website for public comments. Comments were received and responded to until January 2007, when final feedback was received from the ACM. The writing committee then met in February (phone conference) to discuss the best response to these changes. A face meeting was held in April, and another in May, at which significant progress was reported. However, there were also some issues that needed to be resolved.

At the SIGITE 2007 meeting (Sandestin, FL), it was decided to form a steering committee, consisting of two members of the writing committee (Barry Lunt and Joseph Ekstrom), plus a representative of the ACM and one from the IEEE-CS. Later it was determined that these representatives would be Richard LeBlanc (ACM) and Greg Hislop (IEEE-CS). The steering committee gave significant guidance to the process of completing this document.

The final meeting of the writing committee included the steering committee and representatives from the 2-year curriculum committee and took place in Philadelphia, PA, in February 2008.

2.2 Evaluation of curriculum efforts

We believe that this document represents the best collective thinking of Information Technology educators and professionals. Every reasonable effort has been made to identify interested institutions and educators, and to invite them to participate. All meetings have been open meetings; all SIGITE officers have been freely elected by the members. As this document is published, it is hoped that continual input will be sought and incorporated to provide for a continually current Information Technology curriculum.

Computing Curricula Information Technology Volume



Chapter 3 The Information Technology Discipline

3.1 The emergence of Information Technology as a discipline

Information technology is an enormously vibrant field that emerged at the end of the last century as our society experienced a fundamental change from an industrial society to an “information society.” From its inception just half a century ago, computing has become the defining technology of our age, changing how we live and work. Computers are integral to modern culture and are a primary engine behind much of the world's economic and social change.

By the late 1980's desktop workstations and personal computers had largely replaced time-shared main frames as the dominant computing paradigm in many organizations. However, as the personal computer became more powerful and more connected, it became more complex to administer, and the demand for people who could “make things work” in a networked microcomputer environment escalated. The trend to desktop computing was turned into a revolution with the appearance of Web browsers and the resulting explosion of the World Wide Web. By turning the computer into a *usable* communication device that can access the entire world, Web browsers became the first compelling reason for everyone in society to use a computer. The almost overnight acceptance of the WWW by society at large created a hyper-demand for Web-based content and services, which ignited the explosion in demand for Web content developers and Web masters. As Web sites became more active and interactive, the demand for application developers and especially database developers expanded as well.

The field continues to evolve at an astonishing pace. New technologies are introduced continually, and existing ones become obsolete almost as soon as they appear. The rapid evolution of the discipline has a profound effect on Information Technology education, affecting both content and pedagogy. For example, networking was not seen as a major topic area in the early 1990's. The lack of emphasis on networking is not particularly surprising. Networking was not yet a mass-market phenomenon, and the World Wide Web was little more than an idea in the minds of its creators. Today, networking and the Web have become the underpinnings for much of our economy. They have become critical foundations of Information Technology, and it is impossible to imagine that four-year programs would not devote significantly more time to this topic. At the same time, the existence of the Web has changed the nature of the educational process itself. Modern networking technology enhances everyone's ability to communicate and gives people throughout the world unprecedented access to information. In most academic programs today networking technology has become an essential pedagogical tool.

We believe that these changes fall into two categories -- technical and pedagogical -- each of which has a significant effect on Information Technology education. The major changes in both of these categories are described in the individual sections that follow.

3.1.1 Technical changes

Much of the change that affects information technology comes from advances in technology. In the last decade there has been unprecedented innovation in technologies for communication, computation, interactivity, and delivery of information. Over the last ten years the world has changed dramatically both in how people work and live. The technology of telephony and computing has created an increasingly mobile environment where communications and connectivity are expected anytime and anyplace. Society has become accustomed to connectivity that provides access to information on demand in all aspects of every day life. Demand for connectivity to full network service anytime and anyplace has resulted in enormous growth in wireless networks in the last few years comparable to the explosive growth of the Internet in the '90s. Over one billion people have access to the Internet today by wired and wireless access and it is predicted that over two billion people will have wireless access to the Internet by 2010 (Beaulieu, 2002).

Technical advances over the past decade have increased the importance of many curricular topics, such as the following:

- The World Wide Web and its applications
- Networking technologies, particularly those based on TCP/IP
- Systems administration and maintenance
- Graphics and multimedia
- Web systems and technologies
- Service-oriented architecture
- E-commerce technologies
- Relational databases
- Client-server technologies
- Interoperability
- Technology integration and deployment
- Object-oriented event-driven programming
- Sophisticated application programmer interfaces (APIs)
- Human-computer interaction
- Security
- Application domains

In short, it is the advances in computing communication technology, particularly the Internet and the World-Wide Web, which have given rise to the academic field of Information Technology.

3.1.2 Pedagogical changes

The technical changes that have driven the recent expansion of computing have direct implications on the culture of education. Computer networks, for example, make distance education much more feasible, leading to enormous growth in this area. Those networks also make it much easier to share curricular resources among widely distributed institutions. Technology also affects the nature of pedagogy. Demonstration software, computer projection, and individual laboratory stations have made a significant difference in the way Information Technology is taught. The design of Information Technology curricula must take those changing technologies into account.

3.2 The role of Information Technology within the computing disciplines

As an academic discipline, Information Technology focuses on preparing graduates who are concerned with issues related to advocating for users and meeting their needs within an organizational and societal context through the selection, creation, application, integration and administration of computing technologies. An excellent discussion of the Information Technology discipline can be found in the *CC2005: The Overview Report*, page 14:

“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 be 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.”

The diagram in Figure 3.1 is taken from *CC2005: The Overview Report*, as is much of the description of it in this paragraph. *The Overview Report* includes similar diagrams for all 5 computing disciplines, along with more details about this graphical approach to describing them. The shaded area of the diagram represents the focus of typical *information technology* curricula. This area extends down most of the right edge, encompassing the application, deployment, and configuration needs of organizations and people over a wide spectrum of contexts. Across this range (from organizational information systems, to application technologies, and down to systems infrastructure), their role has some overlap with Information Systems, but IT people have a special focus on satisfying human needs that arise from computing technology. In addition, IT’s shaded area goes leftwards, from application towards theory and innovation, especially in the area of application technologies. This is because IT people often develop the web-enabled digital technologies that organizations use for a broad mix of informational purposes, and this implies an appropriate conceptual foundation in relevant principles and theory.

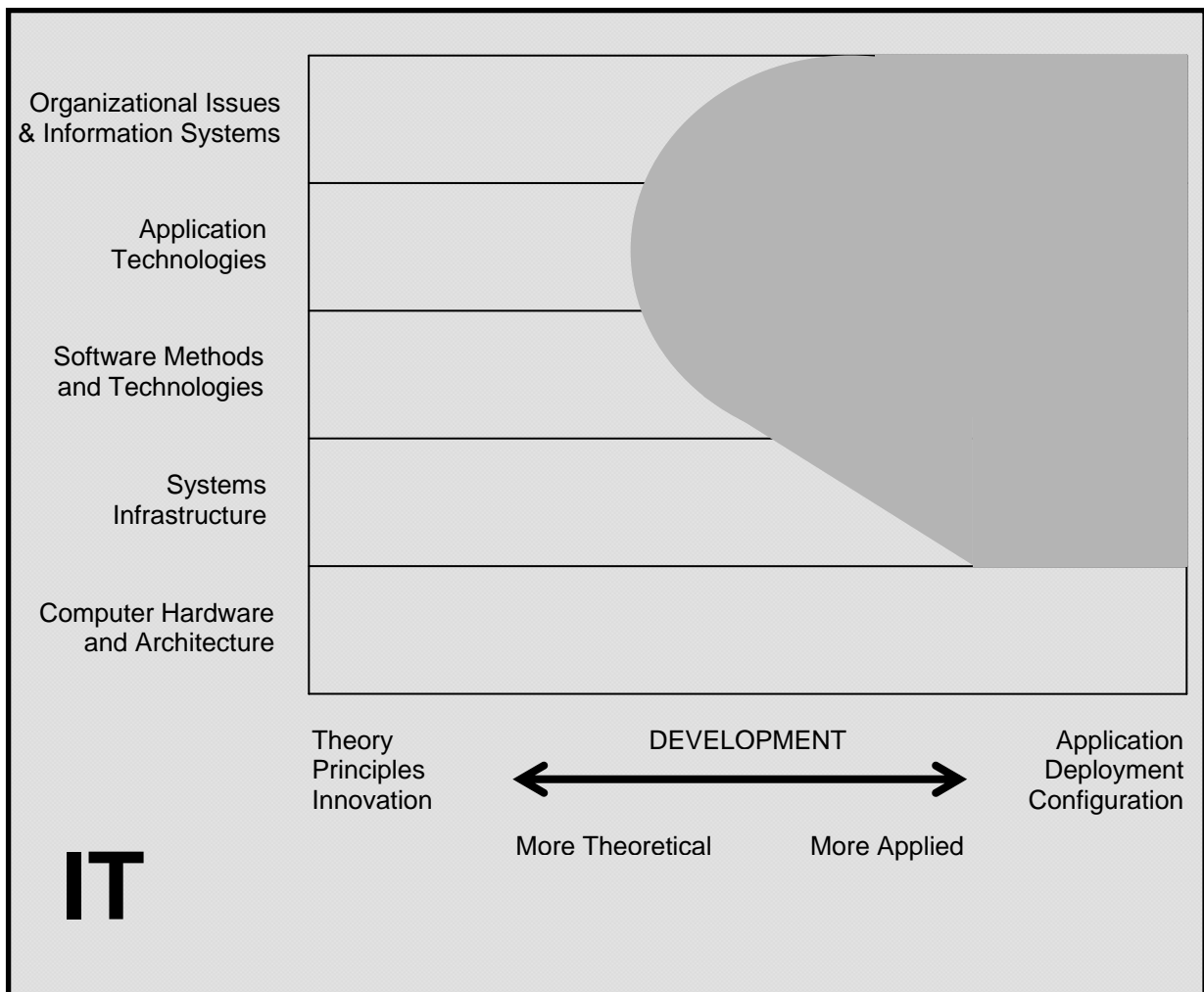


Figure 3-1. Information Technology

3.3 Characteristics of an IT graduate

The fact that Information Technology programs emerged to meet demand from employers has had a significant effect on the evolution of the discipline. Entry-level knowledge and skill requirements gathered from potential employers of graduates naturally translate into learning or program outcomes for graduates from Information Technology programs. In particular, if a graduate is going to be able to function as a user advocate and select, create, apply, integrate and administer computing technologies to meet the needs of users within a societal and organizational context, they need:

- (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
- (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

- (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- (d) An ability to function effectively on teams to accomplish a common goal
- (e) An understanding of professional, ethical, legal, security and social issues and responsibilities
- (f) An ability to communicate effectively with a range of audiences
- (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society
- (h) Recognition of the need for and an ability to engage in continuing professional development
- (i) An ability to use current techniques, skills, and tools necessary for computing practice.
- (j) An ability to use and apply current technical concepts and practices in the core information technologies.
- (k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems.
- (l) An ability to effectively integrate IT-based solutions into the user environment.
- (m) An understanding of best practices and standards and their application.
- (n) An ability to assist in the creation of an effective project plan.

In line with best practices in curriculum design (Sork and Cafarella, 1989; Diamond, 1998), this model curriculum is designed as a blueprint for programs to enable their graduates to achieve these capabilities.

The academic discipline of Information Technology can well be characterized as the most integrative of the computing disciplines. One implication of this characteristic is that a graduate of an IT program should be the first one to take responsibility to resolve a computing need, no matter the source or description of the problem, and no matter the solution that is eventually adopted. The depth of IT lies in its breadth: an IT graduate needs to be broad enough to recognize any computing need and know something about possible solutions. The IT graduate would be the one to select, create or assist to create, apply, integrate, and administer the solution within the application context.

Figure 3-2 depicts the academic discipline of Information Technology. The pillars of IT include programming, networking, human-computer interaction, databases, and web systems, built on a foundation of knowledge of the fundamentals of IT. Overarching the entire foundation and pillars are information assurance and security, and professionalism. While this figure does not depict all aspects of the IT discipline, it does help to describe the relation of the key components.

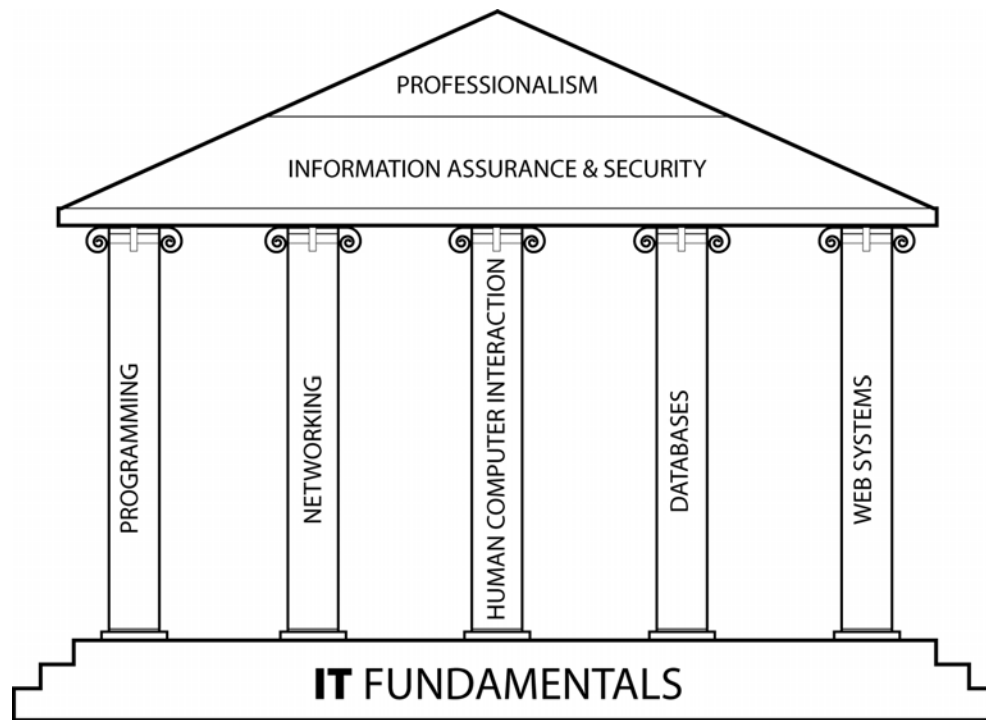


Figure 3-2. The Information Technology Discipline

3.4 Research in IT

Information Technology is the newest computing discipline covered by the Computing Curricula volumes. And, like all the computing disciplines, it is still evolving rapidly. Given this state of affairs, this section presents comments on both the current picture of research in IT and on likely developments in the future.

In addition to the simple newness of the discipline, making definitive statements about research in IT is difficult for several reasons, including:

Focus on practice – IT emphasizes knowledge combined with practical, hands-on expertise. This emphasis is well matched to the challenge of successfully applying information technology in organizational and societal contexts. Most of the IT programs are undergraduate programs. In addition, many of the programs are located at teaching-oriented institutions, perhaps reflecting a greater incentive among these institutions to respond flexibly to career opportunities for graduates.

This history contrasts with disciplinary areas that emerge as research topics first, and then coalesce into disciplines. Development of a research agenda in the IT community is being informed by practice and educational programs. At this early stage in development of the discipline, this may be misinterpreted to conclude that IT lacks sufficient research potential. More accurately, the research agenda simply needs to be viewed as emerging from the practice, and so, by definition, it will develop at a somewhat slower pace.

The computing milieu – *CC2005: The Overview Report (CC 2005)* provides one of the best efforts to date to explain the commonality and difference across a set of computing disciplines. However, it is important to realize that there is a long history of overlap, misunderstanding, and even contention among the disciplines. This is not likely to be resolved any time soon.

This intermingling can be seen along many dimensions that might be expected to separate the disciplines. While some leading journals are clearly tied to particular disciplines, other leading computing publications span multiple disciplines. Similarly, faculty members in one computing department often have research interests tied to another computing discipline.

Even examination of core topics covered by a discipline is not a completely satisfying approach to separating disciplines. In some cases, such as computer engineering, this works reasonably well. But for other cases, even within the disciplines there is active discussion as to the definition of core topics. Reichgelt (2004) discusses this issue with regard to Information Systems.

Given the rapid evolution across the entire landscape of computing, this situation is not surprising, and probably healthy. On the other hand, it greatly complicates the goal of uniquely identifying research by computing discipline.

Terminology – Part of the problem of describing IT research begins with the difficulty of describing computing topics clearly. Terms like “computing” and “information” are badly overworked. Even more focused terms, like “programming”, have meaning from one discipline to another that can vary in ways that are difficult to capture in any short description.

Given these considerations, the following observations seem reasonable in considering IT research:

- As a practice driven discipline, IT has been built on a rich base of existing research. Part of the role of IT is to apply research from the other computing disciplines. Part of the research contribution of IT will be to feed new questions and results back into the research streams on which IT is built.
- Research unique to IT will emerge from the practice of IT. IT research will address questions related to the content of practice, that is, questions about computing. IT research will also address questions related to the process of practice.
- IT research will overlap research in other computing disciplines. All the computing disciplines have overlaps, and IT will be no different. Because IT applies research from the other disciplines, overlap will be even more common.

Discussions within the IT community have resulted in several publications that provide initial ideas about an IT research agenda (Reichgelt, 2004; Ekstrom et al, 2006). The areas identified in these early discussions include:

- **Integration** - Many applications of computing technologies require the integration of different system components (Ekstrom and Lunt, 2003). Viewing systems broadly and including people as components of systems raises a host of integration issues.
- **Trade-off analysis** – Development of IT solutions inherently requires trade-off among approaches, processes, components, etc. Principles and methods for conducting this analysis are needed for successful IT practice.
- **Interface issues** – Integration of system components often results in problems at the interfaces. This is true whether the interfaces involve hardware, or software, or the interface from hardware and software to people.
- **Security and assurance** – Security and information assurance have risen sharply in importance in recent years. Since protection is only as good as the weakest point in the system, security and assurance present particular challenges in IT, where the scope of concern encompasses the total system
- **Implementation** - The introduction of an IT application in a user environment often changes that environment in subtle ways. Since many IT applications are designed for the user environment as it

currently exists, such changes may undermine the ability of the application to meet the needs of users. Being able to predict how an IT application is likely to change the user environment would help ensure successful design.

The list above is intended to capture some flavor of the topics that have been identified as relevant to an IT research agenda. As IT evolves as an academic discipline, topics like these will flow from the unique focus IT has on meeting the needs of users in a societal and organization context through the selection, creation, application, integration and administration of computing technologies.



Computing Curricula Information Technology Volume

Chapter 4 Principles

In formulating this document, the working group followed the following principles:

1. *Although this document can in principle be used as a stand-alone document, the formulation of the curriculum was governed by the desire to provide a blueprint to create accreditable programs.* ABET now offers accreditation in Information Technology; we believe that any program that follows this model curriculum will enable students to achieve the program outcomes mentioned in the ABET CAC accreditation criteria and will meet both the general and the IT specific curriculum criteria.
2. *This document is intended to exist as part of the CC2005 series.* We therefore followed the format developed in other documents within this series, in particular CC2001 – Computer Science, and adopted the terminology in that document to describe the IT body of knowledge. In particular, the IT body of knowledge is organized hierarchically into three levels. The highest level of the hierarchy is the **knowledge area**, which represents a particular disciplinary sub-field. The knowledge areas are broken down into smaller divisions called **units**, which represent individual themes within an area. The units are defined in terms of a set of **topics**; these topics are informed by the learning outcomes for these topics.
3. *Despite the rapidly evolving nature of information technology, we wanted to formulate a curriculum with some longevity.* In formulating knowledge areas for Information Technology, we developed learning outcomes first and allowed topics to follow from the learning outcomes. Outcomes describe skills that are to some extent independent of the technological areas in which the skills are deployed, and therefore have a longer shelf life. Nevertheless, we recommend that the professional associations in information technology establish an ongoing review process that allows individual components of the curriculum recommendations to be updated on a recurring basis.
4. *The curriculum must be flexible and the required body of knowledge must be as small as possible.* There are a large number of careers that graduates from IT programs enter. Those careers show an enormous diversity and the knowledge base and skill sets required for each consequently vary widely as well. The curriculum was therefore designed in a way that gives an institution considerable freedom in tailoring the curriculum to the needs of its students and other institutional stakeholders. For this purpose, we recommend core outcomes that must be met, and provide examples of advanced learning outcomes for additional depth in each unit of each knowledge area.
5. *The curriculum must reflect those aspects that set Information Technology apart from other computing disciplines.* The integration of different technologies and the integration of technologies into organizations are fundamental to Information Technology. An IT graduate must therefore acquire a skill set that enables him or her to successfully perform integrative tasks, including user advocacy skills, the ability to address information assurance and security concerns, the ability to manage complexity through abstraction, extensive capabilities for

problem solving across a range of integrated information and communication technologies, adaptability, outstanding interpersonal skills, high ethical standards, and professional responsibility. The curriculum must reflect these pervasive themes, which are discussed further in Chapters 7, 8 and 10.

6. *The curriculum must reflect the relationship of Information Technology to other computing disciplines.* We recognize that there is a significant overlap between different computing disciplines. Where possible, this model curriculum therefore used knowledge units from existing model curriculum documents.

7. *This document is aimed at four-year programs offered at U.S. institutions of higher learning, but should also be applicable in other contexts.* Despite the fact that curricular requirements differ from country to country, this document is intended to be useful to computing educators throughout the world. Although it has been strongly influenced by educational practice in the United States, we have made every effort to ensure that the curriculum recommendations are sensitive to national and cultural differences so that they will be internationally applicable. Furthermore, although there are distinct differences between four-year programs and other types of programs, we expect aspects of this document to be applicable in other programs.

8. *The development of this volume must be broadly based.* To be successful, the process of creating the recommendations must include participation from many different constituencies including industry, government, agencies involved in the creation of accreditation criteria and model curricula, and the full range of higher educational institutions involved in IT education.

9. *This volume must go beyond knowledge areas to offer significant guidance in terms of implementation of the curriculum.* Although it is important for this volume to articulate a broad vision of IT education, the success of any curriculum depends heavily on implementation details. In particular, although the knowledge-area structure used in other CC volumes can serve as a useful framework, most institutions need more detailed guidance. For such institutions, this volume will be effective only if it defines a small set of example models that assemble the knowledge units into reasonable, easily implemented courses. This volume must also provide institutions with advice on the practical concerns of setting up a curriculum by including sections on strategy and tactics along with technical descriptions of the curricular material.

Computing Curricula Information Technology Volume



Chapter 5 Overview of the IT Body of Knowledge

In developing a curriculum for four-year study in Information Technology, one of the first steps is to identify and organize the material that would be appropriate for that level. As noted in [Chapter 1](#), we sought to accomplish this goal by convening a set of knowledge area focus groups and assigning to each one the responsibility of defining the body of knowledge associated with one of the following knowledge areas:

ITF	Information Technology Fundamentals
HCI	Human Computer Interaction
IAS	Information Assurance and Security
IM	Information Management
IPT	Integrative Programming and Technologies
MS	Math and Statistics for IT
NET	Networking
PF	Programming Fundamentals
PT	Platform Technologies
SA	Systems Administration and Maintenance
SIA	System Integration & Architecture
SP	Social and Professional Issues
WS	Web Systems and Technologies

5.1 Structure of the body of knowledge

As discussed in principle 2 of [Chapter 4](#), the IT body of knowledge is organized hierarchically into three levels. The highest level of the hierarchy is the **knowledge area**, which represents a particular disciplinary subfield. Each knowledge area is identified by a two-letter or three-letter abbreviation, such as PF for *programming fundamentals* or SIA for *system integration & architecture*. The knowledge areas are broken down into smaller divisions called **units**, which represent individual themes within an area. Each unit is identified by adding a numeric suffix to the area name; as an example, **NET2** is a unit on *Routing and Switching*. Each unit is further subdivided into a set of **topics**, which are the lowest level of the hierarchy. The top two levels are shown on the one-page summary of the [IT body of knowledge](#); all three levels are given in [Appendix A](#).

5.1.1 Core and advanced outcomes

As discussed in [Chapter 4](#), one of our goals in proposing curricular recommendations is to keep the required component of the body of knowledge as small as possible. To implement this principle, we made a distinction within knowledge units between **core learning outcomes** and **advanced learning outcomes**. Core learning

outcomes are skills that anyone obtaining a four-year degree in the field must acquire. The core has been limited to those items having broad support as essential for all IT students. Advanced learning outcomes are skills that reflect expectations for units where students do advanced work. Any degree program will require students to achieve some subset of the advanced outcomes. The core and advanced outcomes associated with a knowledge unit typically cover the same topics, but the depth to which the topic is covered and the skill levels that students are expected to achieve differ significantly between core and advanced learning outcomes.

In discussing the IT 2008 recommendations during their development, we have found it helpful to emphasize the following points:

- *The core learning outcomes refers to those skills that all students in all Information Technology degree programs should achieve.* Several topics and learning outcomes that are important in the education of many students are not included in the core. This lack of inclusion in the core does not imply a negative judgment about the value, importance, or relevance of those topics. Rather, it simply means that there was not a broad consensus that the topic should be required of *every* student in *every* Information Technology degree program.
- *The core learning outcomes are not a complete curriculum.* Because the core is defined as minimal, it does not, by itself, constitute a complete four-year curriculum.
- *The core learning outcomes must be supplemented by advanced learning outcomes.* Every four-year program must include material relevant to advanced learning outcomes from the body of knowledge. The IT2008 report does not define what those must be, as this additional work can and should vary based on institutional mission, the areas of concentration offered by a given institution, and individual student choice.
- *Core learning outcomes are not necessarily achieved in a set of introductory courses early in the four-year curriculum.* Although many of the skills defined as core are indeed introductory, there are also some core learning outcomes that clearly can be achieved only after students have developed significant background in the field. For example, we believe that all students must create a significant system, including hardware and software installation and testing, at some point during their four-year program. The material that is essential to successful management of projects at this scale is therefore part of the core, since it is required of all students. At the same time, the project course experience is very likely to come toward the end of a student's four-year program. Similarly, introductory courses may include material relevant to advanced learning outcomes alongside the material relevant to core learning outcomes. The designation *core* simply means *required* and says nothing about the level of the course in which it appears.
- *Overlap can and sometimes should exist between knowledge areas or units.* The concept of **pervasive themes** relates to this and is covered in 7.2.1 and in Chapter 10. These themes will occur many times throughout the curriculum. There are other topics which, while not as recurrent as pervasive themes, could be addressed multiple times from different perspectives. We acknowledge the existence of overlap, and are of the opinion that it is not only necessary but valuable.

5.1.2 Assessing the time required to cover a unit

To give readers a sense of the time required to cover a particular unit, this report follows the lead of the CC2001 task force in choosing to express time in **hours**, corresponding to the in-class time required to present the material in a traditional lecture-oriented format. To dispel any potential confusion, however, it is important to underscore the following observations about the use of lecture hours as a measure:

- *We do not seek to endorse the lecture format.* Even though we have used a metric with its roots in a classical, lecture-oriented form, we believe, as did the CC2001 task force, that there are other styles –

particularly given recent improvements in educational technology -- that can be at least as effective. For some of these styles, the notion of *hours* may be difficult to apply. Even so, the time specifications should at least serve as a comparative measure, in the sense that a 5-hour unit will presumably take roughly five times as much time to cover as a 1-hour unit, independent of the teaching style.

- *The hours specified do not include time spent outside of class.* The time assigned to a unit does not include the instructor's preparation time or the time students spend outside of class. As a general guideline, the amount of out-of-class work is approximately three times the in-class time. Thus, a unit that is listed as requiring 3 hours typically entails a total of 12 hours (3 in class and 9 outside).
- *The hours listed for a unit represent a minimum level of coverage.* The time measurements we have assigned for each unit should be interpreted as the *minimum* amount of time necessary to enable a student to achieve the learning outcomes for that unit. It is always appropriate to spend more time on a unit than the recommended minimum.

5.1.3 Packaging units into courses

The structure and format of courses vary significantly from institution to institution and from country to country. Even within the United States, some colleges and universities use a semester system while others follow a shorter quarter system. Under either system, there can be differences in the number of weeks in a semester, the number of lectures in a week, and the number of minutes in a lecture. We would also like to emphasize that the dynamic nature of the technology involved in information technology will necessitate continual revision and analysis of coverage and placement of the outcomes of the curriculum.

Appendix B includes two sample ways in which the IT body of knowledge could be packaged into courses.

5.2 Summary of the IT body of knowledge

A summary of the IT body of knowledge -- showing the knowledge areas, units, and the minimum time required for each -- appears below. Core units can be identified by having suggested hours of coverage shown in parentheses after the unit title. The details of the body of knowledge appear in [Appendix A](#).

The Information Technology Body of Knowledge

ITF. Information Technology Fundamentals (25 core hours)

- ITF. Pervasive Themes in IT (17)
- ITF. History of Information Technology (3)
- ITF. IT and Its Related and Informing Disciplines (3)
- ITF. Application Domains (2)

HCI. Human Computer Interaction (20 core hours)

- HCI. Human Factors (6)
- HCI. HCI Aspects of Application Domains (3)
- HCI. Human-Centered Evaluation (3)
- HCI. Developing Effective Interfaces (3)
- HCI. Accessibility (2)
- HCI. Emerging Technologies (2)
- HCI. Human-Centered Computing (1)

IAS. Information Assurance and Security (23 core hours)

- IAS. Fundamental Aspects (3)
- IAS. Security Mechanisms (Countermeasures) (5)
- IAS. Operational Issues (3)
- IAS. Policy (3)
- IAS. Attacks (2)
- IAS. Security Domains (2)
- IAS. Forensics (1)
- IAS. Information States (1)
- IAS. Security Services (1)
- IAS. Threat Analysis Model (1)
- IAS. Vulnerabilities (1)

IM. Information Management (34 core hours)

- IM. IM Concepts and Fundamentals (8)
- IM. Database Query Languages (9)
- IM. Data Organization Architecture (7)
- IM. Data Modeling (6)
- IM. Managing the Database Environment (3)
- IM. Special-Purpose Databases (1)

IPT. Integrative Programming & Technologies (23 core hrs)

- IPT. Intersystems Communications (5)
- IPT. Data Mapping and Exchange (4)
- IPT. Integrative Coding (4)
- IPT. Scripting Techniques (4)
- IPT. Software Security Practices (4)
- IPT. Miscellaneous Issues (1)
- IPT. Overview of Programming Languages (1)

MS. Math and Statistics for IT (38 core hours)

- MS. Basic Logic (10)
- MS. Discrete Probability (6)
- MS. Functions, Relations and Sets (6)
- MS. Hypothesis Testing (5)
- MS. Sampling and Descriptive Statistics (5)
- MS. Graphs and Trees (4)
- MS. Application of Math & Statistics to IT (2)

NET. Networking (22 core hours)

- NET. Foundations of Networking (3)
- NET. Routing and Switching (8)
- NET. Physical Layer (6)
- NET. Security (2)
- NET. Network Management (2)
- NET. Application Areas (1)

PF. Programming Fundamentals (38 core hours)

- PF. Fundamental Data Structures (10)
- PF. Fundamental Programming Constructs (10)
- PF. Object-Oriented Programming (9)
- PF. Algorithms and Problem-Solving (6)
- PF. Event-Driven Programming (3)

PT. Platform Technologies (14 core hours)

- PT. Operating Systems (10)
- PT. Architecture and Organization (3)
- PT. Computing Infrastructures (1)
- PT. Enterprise Deployment Software
- PT. Firmware
- PT. Hardware

SA. System Administration and Maintenance (11 core hours)

- SA. Operating Systems (4)
- SA. Applications (3)
- SA. Administrative Activities (2)
- SA. Administrative Domains (2)

SIA. System Integration and Architecture (21 core hours)

- SIA. Requirements (6)
- SIA. Acquisition and Sourcing (4)
- SIA. Integration and Deployment (3)
- SIA. Project Management (3)
- SIA. Testing and Quality Assurance (3)
- SIA. Organizational Context (1)
- SIA. Architecture (1)

SP. Social and Professional Issues (23 core hours)

- SP. Professional Communications (5)
- SP. Teamwork Concepts and Issues (5)
- SP. Social Context of Computing (3)
- SP. Intellectual Property (2)
- SP. Legal Issues in Computing (2)
- SP. Organizational Context (2)
- SP. Professional and Ethical Issues and Responsibilities (2)
- SP. History of Computing (1)
- SP. Privacy and Civil Liberties (1)

WS. Web Systems and Technologies (22 core hours)

- WS. Web Technologies (10)
- WS. Information Architecture (4)
- WS. Digital Media (3)
- WS. Web Development (3)
- WS. Vulnerabilities (2)
- WS. Social Software

Total Hours: 314

Notes:

1. Order of Knowledge Areas: Fundamentals first, then ordered alphabetically.
2. Order of Units under each Knowledge Area: Fundamentals first (if present), then ordered by number of core hours

Figure 5-1. IT body of knowledge with core topics underlined



Computing Curricula Information Technology Volume

Chapter 6 Overview of the Curricular Models

The [body of knowledge](#) provided in [Chapter 5](#) does not by itself constitute a curriculum. To be useful this report will present sample detailed course implementations and strategies for developing individual courses to complete a four-year curriculum. This chapter presents a brief description of the philosophy behind the proposed curricular models. The descriptions for the courses themselves appear in [Appendix B](#).

6.1 Overall structure of the model curricula

The courses described in this report are divided into three categories according to the level at which they occur. Courses designated as *introductory* are intended to be offered within the first or second year of a curriculum. Courses designated as *intermediate* are intended to be offered in the second or third year of a curriculum. Courses designated as *advanced* are intended to be offered late in the curriculum and require knowledge and skills obtained earlier in the curriculum.

It must be noted that the categories of introductory, intermediate and advanced are defined independent of core and advanced outcomes. Core outcomes refer to the body of knowledge and should be offered at the appropriate level in the curriculum.

6.2 The role of experiential learning

In [Chapter 1](#), we stated that IT four-year programs aim to provide their graduates with the skills and knowledge to take on appropriate professional positions in Information Technology on graduation and grow into leadership positions or pursue research or graduate studies in the field. IT professionals are primarily entrusted with the integration of different technologies, and the integration of the technology into organizations. This requires a familiarity with the technology that goes beyond the purely theoretical. IT four-year programs must therefore be designed in a way that allows graduates to develop a practical understanding of the technology. This must not be understood to mean that theoretical knowledge is irrelevant to the IT professional. In fact, without a solid understanding of the underlying theories and concepts, it is unlikely that an IT professional can remain abreast and understand the latest technical innovations. However, just as practical knowledge without a good grasp of the underlying theory is likely to lead to a person whose technical skills will rapidly become obsolete, a graduate from an IT four-year program who fully understands the theory behind a particular technology, but is unable to apply the technology in a practical sense to address the needs of an organization, is likely to be of limited value to that organization.

The six broad goals (section 1.4) and the fourteen program outcomes of IT programs (section 3.3 and Chapter 10) include such words as *apply*, *employ*, *use*, *integrate*, *demonstrate*, and *accomplish*, all of which include a strong application component. While we do not wish to endorse a particular delivery mechanism, we are of the strong opinion that students are unlikely to acquire the practical knowledge described in the

learning outcomes without a significant *experiential learning* component in their program of study. Experiential learning should therefore permeate the IT curriculum.

There are, of course, different ways of providing experiential learning, including but not limited to:

- Instructor demonstrations,
- Structured and unstructured labs,
- Relevant field trips,
- Multi-stage individual and group projects,
- Interviews with IT professionals and/or job shadowing,
- Design, implementation, and documentation projects,
- Preparation and presentation of a technical report,
- Internships and co-ops
- Service learning.

This list exemplifies that experiential learning necessarily goes beyond typing-at-the-keyboard experiences.

Different experiences are appropriate for different learning outcomes. However, in general, we are of the view that the most appropriate delivery of the curriculum proposed in this document requires a mix of the various ways of providing experiential learning. The appropriate mix depends on the institution and the emphases in its IT program.

6.3 Overview of the implementation strategies

Most IT curricula are implemented using one of two common implementation strategies. These strategies, introduced here and described further in Chapter 7 and Appendix B, are *integration first* and *pillars first*. These two approaches are outlined in the two subsections that follow. Each description includes a list of course names, arranged roughly in the order that a student would take them during the first three years of a four-year program. The first two courses in each list are the same, so the discussion of their role will be deferred to the next chapter in the interest of focusing here on the differences between the two approaches. Similarly, fourth-year courses are not included here, since the options possible at this advanced level are not dependent on a choice between the integration-first and pillars-first models.

Details about the courses used in example implementations of each of these strategies are given in [Appendix B](#), including course descriptions, knowledge area and units included, and program outcomes addressed. In the examples, the course numbers have been chosen to indicate the years in which courses are likely to be offered. For example, numbers starting with one are first year courses, numbers starting with 2 are second year courses, etc. The numbers given are only approximate, though they do provide a suggestion. For example, Computing Platforms (in the Integration-First Approach) could also be taught in the second year of the program.

6.3.1 Integration-first approach

The integration-first approach is designed to give students an early integrated view of basic topics across the knowledge areas corresponding to the IT pillars by presenting them together in two key courses during the second year of the curriculum, the IT Systems and Web Systems courses. The IT courses included in the Integration-First example implementation in Appendix B are:

- IT Fundamentals
- Programming Fundamentals
- Computing Platforms

- IT Systems
- Web Systems
- Networking
- Databases
- Human-Computer Interaction
- Information Assurance and Security

Note that the Networking, Databases and Human Computer Interaction courses build upon the introduction to the basic topics in these areas included in the IT Systems course and can all be designed assuming that students have an integrated perspective on the role of these areas in IT systems.

6.3.2 Pillars-first approach

The pillars-first approach introduces the detail of the IT pillars first and provides the integration later in the curriculum. The course titles for the corresponding example implementation in Appendix B have a close correspondence to the components seen in the IT discipline diagram in Figure 3-2. The courses include:

- IT Fundamentals
- Programming Fundamentals
- Fundamentals of Networking
- Fundamentals of Web Systems
- Fundamentals of Information Management
- Fundamentals of Human-Computer Interaction
- System Administration and Maintenance
- Integrative Programming
- Information Assurance and Security

There are courses in this list corresponding to each of the pillars, giving students a detailed, complete view of each of these knowledge areas on its own. The integrated view that is an important aspect of IT thinking is developed in the latter courses in this list, in which instructors can assume that students have a thorough grasp of the pillars material.



Computing Curricula Information Technology Volume

Chapter 7 The Core in the Curriculum

The discussion in section 5.1.1 treats the relationship between core and advanced outcomes, pervasive themes and our approach to their presentation in the body of knowledge. In the following discussion it must be remembered that the term “core” does not refer to the material which must be covered in the introductory courses, but is intended to describe the subset of the body of knowledge which is common to all IT programs, regardless of where that material is presented in the curriculum. A complete program of study must also include advanced material, a subset of which will be chosen based upon institutional focus. In addition, pervasive themes such as security must be treated in their relationships to all of the other areas of study. There are outcomes that are “core” that cannot be achieved until the student has a relatively deep understanding of many other “core” topics. For example, one must have a good understanding of database query, CGI and the generation of dynamic web pages to understand SQL insertion attacks, yet it is a core concept that one has to protect against such embedded syntax in fields passed to the database tier.

This chapter looks at the core material of the IT four-year curriculum and discusses when that material is best covered in the curriculum. Section 7.1 outlines our overall philosophy concerning the introductory curriculum. Section 7.2 looks at the topics that are central to the design of introductory courses. Section 7.3 looks at presentation of the core using the integration-first approach, section 7.4 considers the alternative pillars-first approach, and section 7.5 describes the topics addressed at the advanced level, regardless of the structure of a curriculum.

7.1 Overall philosophy

Information Technology is very much an integrative discipline; it pulls together the IT pillars of databases, human-computer interaction, networking, programming, and web systems and uses a solid background in each of them to enable graduates to solve all types of computing and informational problems, regardless of their origin. As a discipline, IT emphasizes the pervasive themes of user centeredness and advocacy, information assurance and security, and the management of complexity through abstraction and modeling, best practices, patterns, standards, and the use of appropriate tools.

Because of the dynamic nature of computing, the set of available computing majors is usually in flux and it can be hard for students to understand their differences and similarities. It should be the object of all IT programs to give first-year students as much help as possible in choosing the computing major that matches their interests. First-year students would benefit from an introduction to the material found in *CC2005: The Overview Report*, from exposure to faculty who have practiced computing in industry, and from discussions centered around the computing disciplines and their differences and similarities.

Vendor and industry certifications, while intrinsically valuable, are not part of the model curriculum, due to their focus on vendor-specific technologies. More detail on this issue is included in section 9.6.

7.2 Expectations of the introductory curriculum

Depending on the model chosen for presenting the introductory material (integration first versus pillars first; see [Appendix B](#)), students will have different capabilities and different theoretical backgrounds at the end of the first two years. This is particularly significant for institutions with 2-year programs or 2+2 programs.

Regardless of the model chosen, there are some topics which must be covered in the introductory curriculum; these topics include IT fundamentals, programming fundamentals, and the IT pervasive themes.

7.2.1 IT fundamentals

We believe that the introductory sequence (see Appendix B) is best covered in the first year of an IT program. One of the key pieces of this first portion is the IT fundamentals, covering topics such as the nature of IT and its closely-related sister disciplines, pervasive IT themes (see below), the history of IT, and organizational issues that relate to IT. Students who are exposed to this material early in their academic experience (preferably beginning their first term) are well prepared to either continue in the IT program, or decide which computing program (if any) is best for them. We believe the IT fundamentals knowledge area is best taught by someone with an IT focus and practical experience.

7.2.2 Programming fundamentals

Programming is an essential skill for IT students because programming concepts are used in nearly all core courses. Beginning students are commonly exposed to the concepts in the Programming Fundamentals (PF) knowledge area early in the program. Typically, an introductory programming course is included in the first year of an IT curriculum. Both of the curricular approaches presented in Appendix B include a course designated as Programming Fundamentals, which includes all of the units from the PF area.

While the number of hours associated with the units in this area will fit in a typical one semester course, it is not necessarily the case that a student with no previous exposure to programming will be able to achieve all of the expected learning outcomes for these units. While most introductory programming courses taught in IT and other computing departments do not require programming experience as a prerequisite, the reality of high failure and drop rates in these courses suggests that there is something wrong with the common assumption that prior experience does not matter. Programming courses require a student to master a set of concepts and also to develop sophisticated skills. The introductory courses in most other disciplines are highly concept-oriented, making prior preparation a less significant factor. Another discipline with a high skill load is mathematics. Students can begin studying mathematics at a variety of different entry points, with placement depending on their level of preparation. It seems only logical that there should be more than one entry point for the study of programming, which is at least as skill oriented as mathematics.

Following the above reasoning, a two-course programming sequence could be designed starting with a limited agenda of concepts in the first course. These might include most of the topics in the Fundamental Programming Constructs (PF2) unit along with some of the topics from Fundamental Data Structures (PF1) and Algorithms and Problem Solving (PF4). Limiting the demands for concept learning in this course will allow time for substantial work by students on developing programming skills. The second course, designed under the assumption that students have basic programming skills, could then cover the rest of the Programming Fundamentals topics and also include material to provide a bridge to other areas, such as integrative programming, security and the interface programming parts of human-computer interaction. An alternative for programs that include an early practicum experience would be to include practicum-preparation material in the second course. An important aspect of any two-course approach would be to

allow students who can demonstrate adequate preparation via their high school transcripts or a local placement exam to begin in the second course.

As noted at the beginning of this discussion, the two curricular models in Appendix B are defined with only a single Programming Fundamentals course. They are equally amenable to modification to include a second course. IT program designers will want to consider local factors to determine the more appropriate of these two approaches. The two-course approach is particularly intended to produce more uniform outcomes among the students in the second course, which can potentially have a positive impact on subsequent courses, as well.

7.2.3 Pervasive themes

Throughout the lengthy deliberations that went into the preparation of this volume, several topics have emerged that were considered essential, but that did not seem to belong in a single specific knowledge area or unit. We are of the opinion that these topics are best addressed multiple times in multiple classes, beginning in the IT fundamentals class and woven like threads throughout the tapestry of the IT curriculum. These topics, referred to as **pervasive themes**, are:

- user centeredness and advocacy
- information assurance and security
- the ability to manage complexity through abstraction & modeling, best practices, patterns, standards, and the use of appropriate tools
- extensive capabilities for problem solving across a range of information and communication technologies and their associated tools
- adaptability
- professionalism (life-long learning, professional development, ethics, responsibility)
- interpersonal skills

While these themes will be explicitly addressed in the IT Fundamentals course, it is essential that they also be reinforced throughout the curriculum. The sample course sequences in Appendix B illustrate how this could be achieved by indicating the courses in which these themes would be addressed.

7.3 Presenting the core using the integration-first approach

As discussed above, there is no expected difference in how some of the core is presented in these two approaches. This applies specifically to the IT Fundamentals and Programming Fundamentals. However, the remainder of the core classes in the first three years would be somewhat different.

The integration-first approach has the advantage of giving a context for the material at the time it is taught. It also provides an introduction to the vocabulary and models necessary for students to comprehend and discuss the key concepts in the core of an IT curriculum. However, this approach has the disadvantage of making a common computing core more difficult at institutions with multiple computing programs. It also is not the better approach for allowing students to come in from a 2-year program or transfer out after the first two years.

The integration-first approach includes a Computing Platforms course that would be taught to tie in closely with the material covered in Programming Fundamentals. This background prepares students to take the two courses that are central to the integration-first approach: IT Systems, followed by Web Systems. These classes must be taken before courses that cover the other pillars. The remaining three pillars are presented in three classes: Networking, Databases, and Human-Computer Interaction. The IT Systems and Web Systems

courses thoroughly cover the topics in web systems (one of the five pillars), and also give an introduction to the pillars of networking, databases, and human-computer interaction. The advantage of this approach is that by the time the students take the classes on these latter three pillars, they have a context for all the material being covered.

7.4 Presenting the core using the pillars-first approach

The pillars-first approach has the advantage of being a better approach for articulation with two-year programs. It also makes it easier to identify a common computing core for institutions with multiple computing programs. This approach also gives an early depth in the pillars of the IT curriculum core. The primary disadvantage of the pillars-first approach is that it does not provide an overview to how all the core material of the IT curriculum fits together. It also tends to present the vocabulary and models of each pillar in a more isolated context.

Using this approach, students are first given significant depth in each of the five pillars: programming (Programming Fundamentals), networking (Fundamentals of Networking), web systems (Fundamentals of Web Systems), databases (Fundamentals of Information Management), and human-computer interaction (IT Fundamentals of Human-Computer Interaction). The context for the material in these pillars is provided in subsequent classes, particularly Integrative Programming and Information Assurance and Security. Because there is no course that provides an introductory view of the material in each of the pillars, each of the “Fundamentals” courses listed above must begin at the most elementary topics in the area that it covers. While there are disadvantages to this course structure, it does allow the pillars courses to be designed independently and students may take them in any order.

7.5 The core at the advanced level

The core material that belongs at the advanced level of a 4-year IT program includes: 1) information assurance and security; 2) the social and professional issues of IT; 3) an integrative capstone experience; 4) other advanced materials necessary to fill out the IT curriculum; and 5) the IT pervasive themes. The final year is where the true nature of IT as a profession is presented to the students. During this year all of the loose ends should be clearly tied together for the students. No matter how the core concepts and skills have been presented pedagogically during the first three years, students should be placed in the position to use all of their skills together and to understand that there is no part of a system that can be treated in isolation. Though there are different approaches to bring this integration into the curriculum, we believe that it must be present to make it clear to the students that security should be considered throughout the system life cycle and that the creation of a complex system from a set of components is a multidisciplinary team effort that must be planned and executed in a reasoned, professional manner.

7.5.1 Information Assurance and Security (IAS)

We believe there is some IAS material that is best covered after the students have been fully exposed to the five pillars of an IT curriculum, as well as all the other material at the introductory and intermediate levels. IAS is a very integrative knowledge area, and it is very difficult to teach the advanced concepts and issues of IAS without a significant background in Information Technology. It will be observed that there is an IAS course in the fourth year whether in the integration-first approach or pillars-first approach. In the case of the pillars-first approach, there are no core outcomes specified for the course. What we are trying to indicate is that we believe that an integrative security experience is required in the senior year no matter when the core IAS concepts are covered. Every student should be involved with some of the advanced security outcomes; however, we do not specify which ones a particular program will provide. Some programs might emphasize

the network-oriented advanced outcomes, others risk management or forensics. However, we believe that every student needs some advanced, integrative experience in IAS in the 4th year. IAS is the only knowledge area that is also a pervasive theme. It is not surprising that the description of the weaving of the concepts into the curriculum provides challenges unique to this important area of study and practice.

7.5.2 Social and professional issues of IT

The social and professional issues of IT are not significantly different from and have much in common with the issues relevant to other computing disciplines. Planning and managing an organization's IT infrastructure is a difficult and complex job that requires a solid foundation in applied computing as well as the social and ethical impact of that infrastructure. The goals are difficult and intellectually demanding, and the relationships among them are highly complex because they are often in contention. A classic example is security and usability. IT professionals must make value judgments on a daily basis and therefore students must be provided the skills to accept their roles as future practitioners and begin to grasp the responsibility that they will be asked to bear as professionals and the consequences of their actions.

IT graduates will be required to work closely with professionals from all computing disciplines, as well as people from all backgrounds. IT graduates will be qualified to work anywhere there is an IT infrastructure, whether in health care, government, law enforcement, biology, or an industrial setting. IT professionals must therefore be particularly adept at working with people of many backgrounds and perspectives. They must be aware of cultural issues which have an important impact on such matters as human-computer interaction, teamwork, and related matters. They must be able to recognize when safety critical systems are involved, and when security issues are particularly relevant, and they must be able to react accordingly. They must also have strong ethical convictions and accept responsibility for professional and personal behavior.

These skills are best taught throughout the IT curriculum, rather than at the conclusion only. We therefore do not recommend that this material be postponed until the fourth year, but it is certainly appropriate to cover it either again or in more depth in the fourth year. However a given curriculum accomplishes it, IT graduates must be well trained in this area by the time they complete their fourth year.

7.5.3 The integrative capstone experience

The concept of a capstone type of experience in the last part of the curriculum has gained wide support in academia, particularly in the engineering and engineering technology disciplines, and is gaining support in the computing disciplines. The three common elements of nearly all capstone programs are: 1) students are divided into teams of typically 4 to 8 students each; 2) each team is given a real world project or problem to solve; 3) this project takes many weeks to complete (typically 14 or more). There is much research to support the effectiveness of this experience (see Jones SA, Houghtalen R; Dutta D, et al; Butkus, M. A., and Kelley, M.B; Devon, R., et al.; Catalano, G.D.; and Bohmann, L.J., et al., among many others). Due to the highly integrative nature of the experience, and the need for the students to have covered most of the IT curriculum prior to this experience, this material is best covered as close as possible to the end of the fourth year. In the example curricula, two approaches to the capstone experience have been presented. The one presented in the integration-first approach includes a two-semester sequence that covers much of the SIA content along with many SP aspects where the conceptual knowledge is supplied as it is needed during the capstone experience. The approach illustrated in the pillars-first example uses specific courses to cover the SIA and SP concepts and then uses a single semester for the actual capstone experience. However, both approaches place the major project experience in the 4th year after the students have acquired the core competencies so that they are ready for an integrative project experience with a team of peers.

7.5.4 Other advanced materials

As has been previously mentioned, the IT curriculum core by itself does not constitute a complete curriculum. The remaining materials should include: optional classes in IT to allow further depth; advanced courses in a chosen application domain; and materials to round out the curriculum. The first two are discussed below; the last one (materials to round out the curriculum) is discussed in [Chapter 8](#).

7.5.4.1 Optional IT classes

The core by itself constitutes approximately 314 hours of instruction, which in a 4-year program of approximately 1,800 hours of instruction, constitutes only about 18%. For optional IT classes, we recommend courses that cover the core material in greater depth, and courses that cover the IT field in greater breadth, as deemed appropriate to each institution, so that the IT major is as complete as all other majors at a given institution.

7.5.4.2 Application domain classes

The field of IT is broad enough that it has applications in nearly all human endeavors. For example, particular mention has previously been made of the areas of health care, online education, biological sciences, government, and entertainment. We believe that students are much better prepared for employment if they select an application domain prior to completion of their IT curriculum, and then take appropriate courses from that application domain as part of their curriculum. The number of courses for this emphasis will vary, but we recommend it should be no fewer than two.

7.5.4.3 Professional communications class

In the integration-first approach, the professional communications class (Technical and Professional Communication) is given as a third-year course; in the Pillars-First Approach, the same class is given as a second-year course. This difference is not unique to these approaches, but is a function of where practicum is emphasized in the curriculum. If a program is intended to prepare students to be qualified for a work experience (co-op or internship), this course should be given in the second year; otherwise it may be more practical to fit it into the third or fourth year.

7.5.5 IT pervasive themes

The pervasive themes of the IT curriculum were previously described in section 7.2.3 under the IT fundamentals. They are mentioned again here, because they must be fully covered by the completion of the 4-year curriculum. They must be mentioned often, from the first year through the fourth, since they do not readily fit into a single course, and since they need to be regularly woven into the curriculum in order to become an integral part of it.

7.6 Practicum in the IT curriculum

The issue of when a student should be involved in a practicum (internship) experience is orthogonal to the pedagogical model. However, the educational experiences provided by the two models prepare students for different practical experiences at different stages of the program. For example: a student would be prepared by the database courses in a pillars-first approach to be an apprentice DBA possibly in the 2nd year. A student in an integration-first curriculum might be prepared to function in a more general support role and totally unprepared for a specialized position.

While we recommend a practical component in the curriculum, the placement and nature of that practicum is a function of the approach of the program and the organizational environment that supports it. We can therefore only recommend that care be taken to avoid problems incident to placing students in positions for which they are unprepared.



Computing Curricula Information Technology Volume

Chapter 8 Completing the Curriculum

The primary purpose of [Chapter 7](#) was to outline the material in the model curriculum at the introductory, intermediate and advanced levels. As we have emphasized previously in this report, the Information Technology core does not in itself constitute a complete curriculum. To complete the curriculum, IT programs must also ensure that students have the background knowledge and skills they need to succeed as well as the chance to do advanced work that goes beyond the boundaries of the core. This chapter offers strategies and guidelines in each of these areas. Section 8.1 describes a set of general requirements that support the broad education of IT students. Section 8.2 outlines a set of advanced courses to provide depth in the curriculum.

8.1 General requirements

A successful IT graduate needs many skills beyond the technical ones found in the IT body of knowledge. For example, IT students must have a certain level of mathematical sophistication, familiarity with the methods of science, a sense of how computing is applied in practice, effective communication skills, preparation for being a well-rounded and effective member of society, and the ability to work productively in teams. This chapter outlines several general recommendations for IT programs seeking to meet these goals.

8.1.1 The scientific method

The process of abstraction (data collection, hypothesis formation and testing, experimentation, analysis) represents a vital component of logical thought within the field of computing. The scientific method represents a basic methodology for much of the realm of computing, so students should have a solid exposure to this methodology.

To develop a firm understanding of the scientific method, students must have direct hands-on experience with hypothesis formulation, experimental design, hypothesis testing, and data analysis. While a curriculum may provide this experience in various ways, it is vital that students must "do science" -- not just "read about science."

We therefore make the following recommendations about science:

- Students must develop an understanding of the scientific method and experience this mode of inquiry in courses that provide some exposure to laboratory work.
- Students may acquire their scientific perspective in a variety of domains, depending on program outcomes and their area of interest.

8.1.2 Familiarity with application domains

Due to the broad range of application domains for computing in today's society, IT students must be able to work effectively with people from other disciplines. To this end, we recommend that all Information Technology students should engage in an in-depth study of some subject that uses computing in a substantive way.

IT students have a wide range of interests and professional goals. For many students, study of computing together with an application area will be extremely useful. Such work might be accomplished in several ways. One approach is to integrate case studies into IT courses in a way that emphasizes the importance of understanding the application domain. Other approaches might include an extended internship experience or the equivalent of a full semester's work that would count toward a major in that discipline. Additionally, IT students should be encouraged to consider a minor in another discipline. Such opportunities certainly exist in such fields as psychology, sociology, economics, biology, business, or many of the science or engineering disciplines. With some creativity, it is also possible to find applications in areas that might be considered farther afield, often through innovative approaches beyond the scope of a standard Information Technology curriculum.

8.1.3 Communication skills

A widely-heard theme among employers is that IT professionals must be able to communicate effectively with colleagues and clients. Because of the importance of good communication skills in all computing careers, IT students must sharpen their oral and writing skills in a variety of contexts -- both inside and outside of IT courses. In particular, students in IT programs should be able to:

- Communicate ideas effectively in written form
- Make effective oral presentations, both formally and informally
- Understand and offer constructive critiques of the presentations of others
- Have a pleasant demeanor as they work with people on their IT needs, either in person or by phone
- Write appropriate electronic communications (including email, blogs, instant messages, etc.) to all levels of workers in all IT endeavors.

While institutions may adopt different strategies to accomplish these goals, the program for each IT student must include numerous occasions for improving writing and practicing oral communication in a way that emphasizes both speaking and active listening skills.

At a minimum, an IT curriculum should require:

- Course work that emphasizes the mechanics and process of writing
- At least two formal oral presentations to a group
- The opportunity to critique at least two oral presentations

Furthermore, the IT curriculum should integrate writing and verbal discussion consistently in substantive ways. Communication skills should not be seen as separate but should instead be fully incorporated into the IT curriculum and its requirements.

8.1.4 Working in teams

Few computing professionals can expect to work in isolation for very much of the time. Information technology projects are usually implemented by groups of people working together as a team. Information Technology students therefore need to learn about the mechanics and dynamics of effective team

participation as part of their four-year education. Moreover, because the value of working in teams (as well as the difficulties that arise) does not become evident in small-scale projects, students need to engage in team-oriented projects that extend over a reasonably long period of time, possibly a full semester or a significant fraction thereof.

To ensure that students have the opportunity to acquire these skills as undergraduates, we recommend that all IT programs include the following:

- Opportunities to work in teams beginning relatively early in the curriculum.
- A significant project that involves a complex implementation task in which both the design and implementation are undertaken by a small student team. This project is often scheduled for the last year of undergraduate study, where it can serve as a capstone for the undergraduate experience. Some of the details of this experience were discussed previously in Chapter 7.

The experience that students derive from a significant team project can be enhanced further by using teams that cross disciplinary boundaries. As an example, IT students can be paired with students in biology to conduct a project in the emerging area of biocomputation. Such a project will require expertise from both disciplines, along with strategies to support effective communication across the disciplinary boundary. We strongly endorse the concept of multicultural interdisciplinary team projects, and we agree that such projects can provide a rich and valuable experience for students, both inside and outside of information technology.

8.1.5 Becoming a contributing member of society

Regardless of the depth or focus of one's technical background, each person is expected to operate effectively and amicably in society. This includes accepting and valuing the diverse opinions and perspectives of others, awareness that their professional knowledge provides them with unique opportunities to contribute to society and understanding the implications of social and political developments.

8.1.6 Pervasive themes in the IT curriculum

The topic of pervasive themes in the IT curriculum was introduced in [Chapter 7](#). These themes are essential concepts that must be included in the IT curriculum, but don't fit nicely into a single class. One way to ensure that students learn these concepts is to weave them into the fabric of the main curriculum. There is, however, always a danger that elements of the pervasive themes absorb so much time that they overwhelm the primary topics of the curriculum modules in which they are included. At the same time, these pervasive themes are considered essential for IT students, and must be adequately taught. There are delicate issues of balance here, and curriculum and course designers must find the proper mix.

8.2 Advanced courses

We use the term *advanced course* to mean courses whose content addresses advanced learning outcomes. The units in the body of knowledge give testimony to the rich set of possibilities that exist for such courses, but few if any institutions will be able to offer courses covering all the advanced learning outcomes. Institutions will wish to orient advanced courses to their own areas of expertise, guided by the needs of students, the expertise of faculty members, and the needs of the wider community.

A set of potential course titles for each knowledge area appears in Figure 8-1. We have decided not to include in the printed report full descriptions of the advanced courses. Instead, we plan to create web pages for these courses, which will be accessible from the SIGITE web page. By doing so, we will reduce the size of the printed document and, at the same time, allow the documentation associated with each advanced course to remain more up to date.

Figure 8-1. Examples of advanced courses by area

Human-Computer Interaction (HCI) IT301. Human-Centered Design and Evaluation IT302. Graphical User Interface IT303. Multimedia Systems Development IT304. Interactive Systems Development IT305. Computer-Supported Cooperative Work IT306. Human Cognitive Skills	Programming Fundamentals (PF) IT350. Object-Oriented Programming IT351. Event-Driven Programming IT352. Functional Programming IT353. Logic Programming
Information Assurance and Security (IAS) IT311. Cryptography IT312. Forensics and Incident Response IT313. Biometrics IT314. Security Policies and Procedures	Platform Technologies (PT) IT360. Advanced Computer Architecture IT361. Parallel Architectures IT362. Hardware Implementation Technologies IT363. Advanced Computing Techniques
Information Management (IM) IT320. Advanced Databases IT321. Database Design IT322. Transaction Processing IT323. Distributed and Object Database IT324. Data Mining IT325. Data Warehousing IT326. Multimedia Information Systems IT327. Digital Libraries	System Administration and Maintenance (SA) IT370. Network Management IT371. Technical Support IT372. Database Administration
Integrative Programming & Technologies (IPT) IT330. Fundamentals of n-Tier Architectures IT331. Implementing n-Tier Architectures IT332. Advanced Scripting Concepts IT333. Security and the Seams	System Integration and Architecture (SIA) IT410. Software Acquisition and Implementation IT411. System Needs Assessment IT412. Software Economics IT413. Enterprise Systems IT414. Knowledge Management IT415. Computing Economics IT416. Software Testing
Networking (NET) IT340. Advanced Computer Networks IT341. Distributed Systems IT342. Wireless and Mobile Computing IT343. Cluster Computing IT344. Data Compression IT345. Network Security IT346. Enterprise Networking IT347. Digital Communications	Social and Professional Issues (SP) IT420. Professional Practice IT421. Social Context of Computing IT422. Computers and Ethics IT423. IT and Economic Development IT424. Computer Law IT425. Intellectual Property IT426. Privacy and Civil Liberties IT427. Globalization IT428. Change Catalyzed by IT
	Web Systems and Technologies (WS) IT390. Programming for the WWW IT391. E-commerce IT392. Data-Driven Websites IT393. Web Software Tools



Computing Curricula Information Technology Volume

Chapter 9 Professional Practice

As the field of computing continues to change, an unprecedented opportunity exists to make professional practice a seamless part of the curriculum in Information Technology and other computing disciplines. Understanding professional practice is critical for most Information Technology students since the vast majority will enter the workforce upon graduation. In this chapter, we explore various strategies for incorporating professional practice into the Information Technology curriculum. The individual sections review the underlying rationale, current practice in education, support for professional practice from both the private and public sector, techniques for incorporating professional practice into a curriculum, and strategies for assessing the effectiveness of those techniques.

9.1 Rationale

The need to incorporate professional practice into the curriculum is based upon real-world issues, such as the needs of the public and private sector, the public's demand for higher quality products, the increasing number of computing liability cases, and the need to promote life-long learning after graduation. In most cases, students enter school without a complete knowledge or appreciation for these issues, which is a source of frustration both for those who teach these students and for those who hire them. Indeed, as students learn more about professional practice and the underlying issues, they become more interested in their studies and how they can work well with others. Incorporating professional practice into the curriculum can therefore serve as a catalyst to awaken and broaden a student's interest in computing.

Both the private and public sectors have a vested interest in students learning professional practice. They find that students who have experience with the realities of professional work understand the value of interpersonal skills in collaborating with team members and clients, maintain their focus on producing high-quality work, adhere to strong ethical convictions, contribute their time and talents to worthy outside causes, engage in life-long learning, and participate in improvements in their firm. Each year, for example, the National Association of Colleges and Employers conducts a survey to determine what qualities employers consider most important in applicants seeking employment. The following ten factors have been frequently identified:

1. Communication skills (verbal and written)
2. Honesty/integrity
3. Teamwork skills
4. Interpersonal skills
5. Motivation/initiative
6. Strong work ethic
7. Analytical skills
8. Flexibility/adaptability
9. Computer skills

10. Self-confidence

That employers seek candidates with these general qualities underscores the importance of making professional practice a central component of the curriculum.

The growing demand for better, less defect-ridden products has also increased the pressure to incorporate professional practice into the curriculum. Haphazard web-systems design techniques are widely recognized as a significant factor in producing web systems with a high number of defects. As a result, clients are demanding proof of sound software processes before they will sign a contract with a web system provider. Students need to understand the value of establishing face-to-face relationships with clients, agreeing to requirements that can be implemented, and producing the highest quality web systems possible.

Both the IEEE and the ACM promote the development of professional responsibility in several ways.

- They develop and promote codes of ethics [[ACM Code of Ethics](#), [IEEE Code of Ethics](#), [SEEPP Code of Ethics](#)] to which members are expected to adhere. These codes, in general, promote honesty, integrity, maintenance of high standards of quality, leadership, support of the public interest, and life-long learning.
- They sponsor established subgroups -- the Society on Social Implications of Technology (SSIT) and the Special Interest Group on Computers and Society (SIGCAS) -- that focus directly on ethical and professional issues.
- They develop and refine curricular guidelines, such as the ones in this report and its predecessors.
- They participate in the development of accreditation guidelines that ensure the inclusion of professional practice in the curriculum [ABET2008-2009, CSAB2006].
- They support the formation of student chapters which encourage students to develop a mature attitude toward professional practice.
- They provide opportunities for lifelong professional development through technical publications, conferences, and tutorials.

Both students and society must be educated as to what they can and should expect from people professionally trained in the computing disciplines. Students, for example, need to understand the importance of professional conduct on the job and the ramifications of negligence. They also need to recognize that the professional societies, through their codes of ethics and established subgroups emphasizing professional practice, can provide a support network that enables them to stand up for what is ethically right. By laying the groundwork for this support network as part of a four-year program, students can avoid the sense of isolation that young professionals often feel and be well equipped to practice their profession in a mature and ethical way.

9.2 Current practice in education

Many strategies currently exist for incorporating professional practice into the curriculum. Among the most common characteristics of these strategies are courses that help students strengthen their communication, problem-solving, and technical skills. These skills may be fostered in computing courses or, alternatively, in courses outside the Information Technology department, such as a speech class in a communication department or a technical writing class in an English department. Accreditation bodies, however, usually require not only that students *acquire* these skills – either through general education requirements or through courses required specifically for Information Technology – but also that students *apply* these skills in their later courses.

The level of coverage assigned to professional practice varies depending on institutional commitment, departmental resources, and faculty interest. For example, in 1999, Laurie King (Department of Mathematics

and Computer Science at Holy Cross College) conducted an informal survey concerning the inclusion of ethics in curricula through the ACM SIGCSE list. Of the 74 schools that reported back, 40 schools had enough coverage of ethics to meet CSAB Criteria 2000 [CSAB2000]. Although many schools clearly did not consider this material to be essential, it is encouraging that more than half of the schools did. With the growing emphasis on professionalism in accreditation criteria, it is likely that other schools will strengthen their commitment to teaching professional practice.

The following list outlines several potential mechanisms for incorporating additional material on professional practice:

- *Senior capstone courses.* These courses typically form a one- to a two-semester sequence during the student's last year. Usually, students must work in teams to design and implement projects, where those projects must involve consideration of real-world issues including cost, safety, efficiency, and suitability for the intended user. The projects may be developed solely for the class, but may also involve other on- or off-campus clients. Although the emphasis of the course is on project work and student presentations, some material on intellectual property rights, copyrights, patents, law, and ethics may be included.
- *Professionalism, ethics, and law courses.* These courses are one semester long and expose students to issues of professional practice, ethical behavior, and computer law. Topics included may be history of computing, impact of computers on society, computing careers, legal and ethical responsibilities, and the computing profession.
- *Practicum/internship/co-op programs.* These programs are sponsored by the institution (preferably) or department to allow students to have the opportunity to work in industry full- or part-time before graduation. Having adequate administrative support for such programs is essential to their success. Students typically work during the summers and/or from one to three semesters while they are engaged in their four-year degree. The students who do a co-op or internship generally do so off-campus and so may interrupt their education for a summer or a semester. Students are usually paid for their work, but in some cases may also be allowed course credit.
- *Team-based implementation courses.* These courses emphasize the process of IT system development and typically include a team project. Course topics include development processes, project management, economics, risk management, requirements engineering, design, implementation, maintenance, software and hardware retirement, system quality assurance, ethics, and teamwork. Topic coverage is usually broad rather than in-depth.

Many courses outside the Information Technology department can also help students to develop stronger professional practice. Such courses include, but are not limited to, philosophical ethics, psychology, business management, economics, technical communications, and engineering design.

9.3 Supporting professional practice

Support for including more professional practice in the curriculum can come from many sources. The sections that follow look at the responsibilities of the public and private sectors; the relationship between academic preparation and the work environment; and the roles of university administrations, faculty, and students in making professional practice an educational priority.

9.3.1 The private and public sectors

Most students graduating from universities go on to employment in the private or public sector. In their role as the primary consumer of graduating students, industry and government play an important role in helping educational institutions promote professional practice. As an example, students who take advantage of industrial co-ops or government internships may mature faster in their problem-solving skills and become

more serious about their education. Such internships may also help the institutions that offer them, in that a student who has an internship with a company may choose to work there again after graduation. With private/public sector support, professional practice coverage is given a necessary augmentation both inside and outside the classroom.

One of the most important ways that the private and public sectors can support the education process is to encourage their employees to play a greater role in helping to train students. These employees can offer support in a number of ways:

- They can function in the role of mentors to students working on projects.
- They can give special presentations to classes telling students and faculty about their firm, their work, and their development processes.
- They can take part-time positions as adjunct instructors to strengthen a university's course offerings.
- They can provide in-house training materials and/or classes to faculty and students in specialized research, process, or software tool areas.
- They can serve on industrial advisory boards, which service allows them to provide valuable feedback to the department and institution about the strengths and weaknesses of the students.

In each of these ways, institutions in the private and public sectors can establish important lines of communication with the educational institutions that provide them with their future employees.

In addition to the various opportunities that take place on campus, industry and government also contribute to the development of strong professional practice by bringing students and faculty into environments outside of academia. Students and faculty may take field trips to local firms and begin to establish better relationships. Over a longer term, co-op, practicum, and internship opportunities give students a better understanding of what life on the job will be like. In addition, students may become more interested in their studies and use that renewed interest to increase their marketable potential. Students may also form a bond with particular employers and be more likely to return to that firm after graduation. For faculty, consulting opportunities establish a higher level of trust between the faculty member and the company. As a result of these initiatives, employers, students, and faculty know more about each other and are more willing to promote each other's welfare.

In what remains one of the most important forms of support, private and public sectors may also make donations or grants to educational institutions and professional societies in the form of hardware, software, product discounts, money, time, and the like. Often, these donations and grants are critical in providing updated resources, such as lab hardware and software, and in funding student scholarships/awards as well as faculty teaching/research awards. They serve to sponsor student programming, design, and educational contests. Grants can enable more research and projects to be accomplished. At this level, private/public sectors help to ensure the viability/progress of future education and advances in the computing field.

Through patience, long-term commitment, understanding of each other's constraints, and learning each other's value systems, institutions in the private/public sector and in education can work together to produce students skilled in professional practice and behaviors. Their cooperative agreement is essential for producing students who value a high ethical standard and the safety of the people who use the products the students will develop as professionals.

9.3.2 Modeling local and international work environments

Just as industry representatives increasingly seek graduates who are "job ready," most students expect to practice computing in the workplace upon graduation without significant additional training. Although the educational experience differs from that of the workplace, educators need to ease the transition from academia to the business world by:

- Mimicking the computing and networking resources of the work environment
- Teaching students how to work in teams
- Providing significant project experiences

Introducing these points into the curriculum makes it possible to model significant issues in the local and international work environment. Faculty can discuss and have students apply international, intercultural, and workplace issues within the context of computing resources, teamwork, and projects.

Because computing and networking environments change rapidly and several different ones exist, it is not possible to predict the exact environment that students will use upon graduation. As a result, it is not advisable to focus attention in the curriculum on a particular set of tools. Exposure to a wide variety of computing platforms and web system tools provides good preparation for professional work, resulting in flexible learners rather than students who immaturely cling to their one familiar environment.

Learning how to work in teams is not a natural process for many students, but it is nonetheless extremely important. Students should learn to work in both small and large teams so that they acquire planning, budgeting, organizational, and interpersonal skills. Ample course material should support the students in their teamwork. The lecture material may include project scheduling, communication skills, the characteristics of well-functioning and malfunctioning teams, and sources of stress for team environments. Assessment can be based on the result of the team's work, the individual work of the members, or some combination thereof. Team member behavior may also play a factor in the assessment.

Significant project experiences can enhance the problem-solving skills of students by exposing them to problems that are not well defined or that do not have straightforward solutions. Such projects may be a controlled, in-class experience or have a certain amount of unpredictability that occurs with an outside client. The project should serve to stretch the student beyond the typical one-person assignments that exercise basic skills in a knowledge area. Beyond that, projects can also cut across several knowledge areas, thereby helping students to bring all their basic skills together.

9.3.3 Administration, faculty, and student roles

At the highest institutional level, the administration must support faculty professional and departmental development activities. Such activities may include consulting work, professional society and community service, summer fellowships, obtaining certifications and professional licensure, achieving accreditation, forming industrial advisory boards with appropriate charters, establishing co-op/internship/practicum programs for course credit, and creating more liaisons with the private and public sectors. Such activities can be extremely time-consuming. They are, however, enormously valuable to both the individual and the institution, which must take these activities into account in decisions of promotion and tenure.

Faculty and students can work together by jointly adopting, promoting, and enforcing ethical and professional behavior guidelines set by professional societies. Faculty should join professional societies and help to establish student chapters of those societies at their institutions. Through the student chapters, awards may be given for significant achievement in course work, service to the community, or related professional activities. In addition, student chapters may provide a forum for working with potential employers and be instrumental in obtaining donations, speakers, and mentors from outside the institution.

9.4 Incorporating professional practice into the curriculum

The incorporation of professional practice must be a conscious and proactive effort because much of the material must be interwoven into the fabric of existing curricula. For example, the introductory courses in the

major can include discussion and assignments on the impact of computing and the Internet on society and the importance of professional practice. As students progress into their sophomore-level courses, they can start to keep records of their work as a professional might do in the form of requirements, design, and test documents.

Additional material, such as computer history, digital libraries, search techniques, techniques for tackling ill-defined problems, teamwork with individual accountability, real-life ethics issues, standards and guidelines, legal constraints and requirements, and the philosophical basis for ethical arguments, may also be covered either in a dedicated course or distributed throughout the curriculum. The distributed approach has the advantage of presenting this material in the context of a real application area. On the other hand, the distributed approach can be problematic in that professional practice is often minimized in the scramble to find adequate time for the technical material. Projects, however, may provide a natural outlet for much of this material particularly if faculty can recruit external clients needing non-critical systems. When they engage in service-learning projects in the community or work with external clients, students begin to see the necessity for ethical behavior in very different terms. As a result, those students learn much more effectively how to meet the needs of the client's ill-defined problem. No matter how professional practice is integrated into the curriculum, however, it is critical that this material be reinforced with exercises, projects, and exams.

For departments with adequate faculty and resources, courses dedicated to teaching professional practice may be appropriate. These courses include those in professional practice, ethics, and computer law, as well as senior capstone and other appropriate courses. More advanced courses on web system economics, quality, safety, and security may be included as well. These courses may be from disciplines outside of Information Technology and still have a profound effect on the professional development of students.

9.5 Assessing professional practice work

Faculty can promote the positive assessment of professional practice work by establishing an infrastructure where student work is evaluated under common standards and where professional completion of assigned work is actively encouraged. The infrastructure may be built upon the following:

- Outcomes-based assessment
- Reviewing assignments, projects, and exams for appropriate inclusion of professional practice material
- Critically reviewing and establishing sound measurements on student work to show student progress and improvement
- Getting students involved in the review and assessment process so that they get a better sense of the assessment process
- Employing professionals in the private and public sectors to help in assessing student project work
- Using standardized tests to measure overall student progress
- Taking post-graduation surveys of alumni to see how well alumni thought their education prepared them for their careers
- Obtaining accreditation to demonstrate that certain education standards for professional practice have been met

The assessment process should encourage students to employ good technical practice and high standards of integrity. It should discourage students from attempting to complete work without giving themselves enough time or in a haphazard manner, such as starting and barely completing work the night before an assignment is due. The assessment process should hold students accountable on an individual basis even if they work collectively in a team. It should have a consistent set of measurements so that students become accustomed to using them and learn how to associate them with progress or lack thereof.

9.6 Certifications

We acknowledge the value of vendor and industry certifications, and encourage students to pursue them as they see necessary. However, we do not believe that academic credit should be offered for completion of such certifications, nor for training exclusively designed to prepare for these certifications unless it also covers all relevant learning outcomes defined in this document.

Reasons for this position should be articulated. Many institutions offer certification training but do not have regional accreditation. Most certifications are practice-oriented and do not focus on the underlying theories and concepts. Additionally, many certifications are specific to a given vendor and are very narrowly focused. They therefore usually do not meet the learning outcomes defined in this document.

Computing Curricula Information Technology Volume



Chapter 10 Characteristics of IT Graduates

As stated in [Chapter 1](#), four-year IT programs aim to provide their graduates with the skills and knowledge to take on appropriate professional positions in information technology upon graduation and grow into leadership positions or pursue research or graduate studies in the field. Fundamental to information technology is the integration of different technologies and the integration of technologies into organizations. An IT graduate must therefore acquire a skill set that enables him or her to successfully perform integrative tasks, including:

- (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline
- (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- (d) An ability to function effectively on teams to accomplish a common goal
- (e) An understanding of professional, ethical, legal, security and social issues and responsibilities
- (f) An ability to communicate effectively with a range of audiences
- (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society
- (h) Recognition of the need for and an ability to engage in continuing professional development
- (i) An ability to use current techniques, skills, and tools necessary for computing practice
- (j) An ability to use and apply current technical concepts and practices in the core information technologies
- (k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems
- (l) An ability to effectively integrate IT-based solutions into the user environment
- (m) An understanding of best practices and standards and their application
- (n) An ability to assist in the creation of an effective project plan.

As introduced previously in [7.2.1](#), a number of pervasive themes run throughout these program outcomes, namely:

User centeredness and advocacy. IT graduates tend not to design and integrate IT-based solutions for their own sake; rather, they design and integrate IT-based solution to help users and/or organizations achieve their objectives. An integrated IT-based solution includes both technological elements, such as hardware, networking, software and data, as well as people and processes. In order to be successful, IT graduates must therefore develop a mind-set that does not allow losing focus on the importance of users and organizations. They must therefore develop a user-centered approach to technology (HCI, human factors, ergonomics, cognitive psychology, etc.), an awareness of the activities and processes that the solution is expected to support, and a realization that solutions to problems are not always purely technological. Many user and organizational issues can be resolved through other than purely technological solutions, be they additional training or process redesign. (Chumer, 2002)

Information Assurance and Security. IT applications and the data and information stored in such applications are some of the most important assets that an organization possesses. It is crucial that such assets be protected, and security must therefore be a central consideration in any attempt to select, create, integrate, deploy and administer IT systems. While security considerations are important to any computing professional, they become even more important for IT graduates. Security breaches typically occur where different components of a system interface, be it in the interface between different computers in a networked application, or across the interface between the user and the other components of the system. Since IT professionals typically integrate different, often pre-existing components (outsourced or legacy), a lot of their professional activity takes place at such interfaces, and a constant awareness of the possibility of security breaches will therefore enable them to design IT-based solutions that are less likely to put the organization's assets at risk.

The ability to manage complexity through abstraction & modeling, best practices, patterns, standards, and the use of appropriate tools. IT-based solutions are typically designed to address problems or opportunities that arise in a complex environment. Moreover, the integration of an IT-based solution itself often makes an already complex environment even more complicated. IT graduates must be able to handle such complex situations and to focus on those aspects of the situation that are most relevant to the user and the wider context in which the user is expected to function. The most appropriate conceptual tool to deal with complexity is abstraction, and IT graduates therefore must develop the ability to use abstraction to form a model of the situation in which the need for an IT-based solution arises and in which the IT-based solution has to be integrated.

Extensive capabilities for problem solving across a range of integrated information and communication technologies and their associated tools. At the same time, IT graduates must have the skills and knowledge to use the technology appropriately. This requires extensive capabilities in the core information technologies, including programming, web, information management, computer hardware and networking, and HCI. However, IT graduates must also realize that very few IT-based solutions are designed and built from scratch. IT-based solutions are typically constructed from pre-designed components, including legacy applications that the organizations already have in place. IT graduates must therefore be able to use their technical expertise to integrate existing and new technologies.

Adaptability. IT graduates must also be extremely adaptable. The need for adaptability arises partly because of the rapid change in the technology itself. Many of the technologies that are covered in a four-year program are likely to be outdated shortly after graduation. IT graduates must therefore develop life-long learning habits. On the other hand, very few organizations can afford to replace all their technologies wholesale on a regular basis. Many IT applications therefore consist of a hodgepodge of legacy and current and emerging technologies, and the successful IT graduate must be willing not only to become familiar with emerging technologies, but also, if required, with legacy technologies.

Professionalism. IT professionals will be involved at all levels in organizations, and must exhibit the highest levels of professionalism. This pervasive theme includes the sub-themes of life-long learning, professional development, ethics, and responsibility; together with *Interpersonal skills* (see below) these pervasive themes define the face of an IT professional that the organization and general public sees first.

Interpersonal skills. IT-based solutions are developed in teams, consisting of people with different backgrounds, knowledge, skills and values. IT graduates must develop the ability to function effectively in such diverse teams. This requires them to develop superior interpersonal skills, including effective oral, written, presentation and listening skills. Moreover, since IT graduates will often be the interface between users and the technology, they must develop the ability to translate the language of users into technical language, and vice versa. This in turn requires an appreciation of adjacent organizational functions, and an awareness of organizational culture. It also requires respect for and appreciation of

diversity and the ability to tolerate and appreciate different points of views and approaches to problems or opportunities that arise.



Computing Curricula Information Technology Volume

Chapter 11 Computing Across the Curriculum

In the spirit of the CS volume of CC2001, this chapter consists of articulating those general-education courses in Information Technology aimed at students who use computers as a result of the dramatic growth of computing and the enormous impact that computing is making on virtually every field of study. These are Information Technology courses tailored to the very needs and characteristics of the students who are not in Information Technology related majors. These are Information Technology courses “bent to fit user needs” and this is where information technology can play a vital role in developing these courses since it matches very well with the main characteristics of the Information Technology as a discipline which “bends the technology to fit user needs”.

As outlined in CC2001, the [\[CSTB99\]](#) National Research Council report identifies three distinct types of knowledge that are appropriate to consider for inclusion in a general-education course. The following are those distinct types of knowledge areas for inclusion in a general-education course in information technology:

- *Information technology-specific skills:* This class of knowledge refers to the ability to use contemporary Information technology applications such as information management, networking, information assurance, human-computer interaction, and Web systems and technologies.
- *Fundamental and enduring information technology concepts:* Concepts explain the how and why of information technology and they give insight into its opportunities and limitations. They include knowledge areas such as persuasive themes in IT, history of information technology, application domains, organizational issues, data modeling, data organization and retrieving, integrative programming, emerging technologies, and system integration and architecture.
- *General intellectual capabilities:* This class of knowledge areas consists of broad intellectual skills important in virtually every area of study, not simply information technology. These skills allow students to apply information technology to complex tasks in effective and useful ways. Examples include problem solving, managing complexity through abstraction, modeling, use of appropriate tools, inter-personal skills, project management, developing effective interfaces, assets management and cost/benefit analysis, logical reasoning, ethics, and effective oral and written communication skills. These capabilities are beneficial to all students and help to develop and improve a student's overall intellectual ability.

The rest of this chapter outlines the process of course specification, design, implementation, and assessment, and identification and description of three distinct course formats in which Information Technology departments might choose to offer general-education courses in information technology.

11.1 Process questions

A useful model for the course development process is, appropriately enough, the software development process. As with software, the course development process can be divided into four phases: specification, design, implementation, and assessment. We elaborate on each of the phases in the sections that follow.

11.1.1 Course specification

The design of a general-education course entails asking and answering a number of important questions. But to whom should these questions be addressed? Who should have the primary responsibility for specifying the goals and content of a general-education course in information technology? While Information Technology faculty must, of course, be fully involved in helping to formulate specifications, we must be careful not to dictate them. It is important that an in-depth discussion of course goals occur both *inside* and *outside* information technology to ensure that course design is driven by curricular needs and not simply by a desire to teach a certain type of class. In the past, mathematics departments have been criticized for creating introductory courses that focus almost exclusively on pure mathematics, even though many students are interested in and need more applied topics. Information technology should not repeat this mistake. While we should offer assistance during course design, we must also listen carefully to the needs of students and faculty from other departments and be responsive to these needs.

There are four possible goals of a general-education course in information technology:

1. To satisfy general student interest in learning more about information technology
2. To meet institutional distribution requirements in the physical and/or mathematical sciences
3. To give students knowledge of and experience with the effective use of information technology in their own discipline
4. To provide a broader understanding of information technology required for effective participation in society

The first step in developing new general-education courses is identifying a curricular need that is not currently being met. This may be done either reactively or proactively. Information Technology departments should certainly respond to requests from faculty or industry representatives for a new course that could be quite useful to their students or employees. Alternately, Information Technology departments can approach other departments with a proposal for a new course that covers material not included in the existing curriculum. Regardless of how a need is identified, if there is interest expressed by all parties the next step is to identify the target audience and seek input from everyone with a stake in the course's content and structure. A number of questions are appropriate to pose at this time:

- What need will this course meet that is not currently being met by existing courses in the curriculum?
- Who is the target audience for this course? Which departments and programs within the university are likely stakeholders in the course? What type of student will enroll? Do we have some way to measure the interest and demand for such a course? Will the students we are trying to reach have room for this new course within their existing program?
- How will teaching the course affect our own department? Will it have an adverse impact on our ability to teach Information Technology majors?
- How will credit be awarded? Will the course count for general education or distribution credit, major or minor credit in some program(s), or university elective? Or will the course, instead, be offered only as training or continuing education credit?
- Who will teach the course? Will it be team-taught? Who receives credit for developing and teaching it? Do we have sufficient faculty to teach this course even when people are on leave? If not, how can we retrain existing faculty or hire additional faculty with the necessary skills?

In addition to reviewing the responses to these, and similar, questions, departments should carefully read and examine the National Research Council report *Being Fluent with Information Technology* [CSTB99]. This report addresses the fundamental goals and purposes of general-education courses in computing, and it lays an excellent foundation for understanding the issue of computing across the curriculum. This report, along with the responses of client departments to the above questions, will provide the input needed for a detailed course design.

11.1.2 Course design

Once a curricular need has been clearly identified and all departments support the development of a course to meet this need, the next step is course design. Course design involves identifying explicit educational goals and objectives by specifying the technical skills and concepts to be included in the course syllabus and the educational outcomes that we want our students to have. To do so, it is important to pose these basic questions:

- What specific information technology skills should be included in the course, and are these skills important and current to the field of study? What level of expertise in these skills do we want our students to achieve?
- What fundamental and enduring information technology concepts should be included in the course, and how do these concepts relate to and support the information technology skills being taught?
- What social and ethical issues, if any, should be included in the course to complement the technical material being presented?

11.1.3 Course implementation

Once the general course content and goals have been established, developers can turn their attention to implementation-specific details about how the proposed course will be structured by asking themselves the following questions:

- Should the class be taught using a large lecture format or small discussion sections? Should it include a formal laboratory? Informal laboratory? No laboratory?
- What learning activities are most useful for developing specific technical skills? Should there be few large projects? More small projects? Team assignments? What about written papers and/or oral presentations to improve communication skills?
- How can we best evaluate students' learning? What types of projects and/or examinations will be most effective at measuring student success in meeting course goals?
- What instructor expertise is necessary for teaching the course? Do we have such expertise in one individual or would it be better to use a team-teaching approach?
- Do we have adequate educational resources (e.g., computers, laboratories) to offer this course?
- Will there be sufficient student interest to generate adequate enrollment? How often should the course be offered? How many credits should the course be and how many times a week will it meet?

The answers to these and other implementation questions will often be determined not by lofty academic goals but by local concerns and resource constraints that are beyond the scope of this report. These factors could include such issues as enrollment pressures, financial considerations, student populations, college distribution requirements, faculty interests, space limitations, and the working relationship between an Information Technology department and other departments. But, regardless of how they may be answered, a department should be able, based on responses to these questions, to implement a general-education course that goes a long way toward meeting the desired goals.

11.1.4 Course assessment

Following implementation, a department is ready to offer the new general-education course to the college community. This leaves only the final step in the course development process: assessment. After the course has been offered once or twice, its design and implementation should be carefully reviewed and evaluated. The data needed for assessment can be collected in a number of ways: written student evaluations, in-class observations, and personal interviews with students and faculty from the client departments. Once the course has been taught for a few years it is also a good idea to interview graduates regarding the value of this course to their professional work environment.

Some of the questions that should be asked during course assessment include the following:

- Does this course meet its stated goals? If not, should we redesign it or simply eliminate it from the program and consider an alternative approach?
- Has any important topic been omitted? Is anything unnecessarily included?
- Based on examination results and course evaluations, do students completing the course possess the desired skills, knowledge, and capabilities?
- Is the client department satisfied with our course offering? If not, what can we do to improve their satisfaction?

The design and implementation of a general-education course is not a one-time process but rather a "work in progress" that must be updated and modified as we gain additional experience. Course design must include regular reviews and redesign, just as in the software development process. Such reviews are especially important in light of the rapidly changing nature of our field.

11.2 Course models

We have identified three types of courses that can be offered by an Information Technology department: general fluency, area-wide, and single discipline.

- *General fluency:* These courses address skills and concepts that are appropriate for all students at an institution, regardless of their specific field of study. General fluency courses are not concerned with providing specific computer-related skills to a particular discipline. Instead, they are meant to satisfy general student interests in computing, to meet college distribution requirements, and to help produce more informed citizens with respect to information technology. An example of this type of course is IT Fundamentals, given in Appendix B under Integration-First Approach.
- *Area-wide or multidisciplinary courses:* Area-wide courses serve several departments that share a common need for particular computing skills and concepts. They share the characteristic that most, if not all, prerequisite material comes from outside computing. Information Technology departments may work with other departments to identify this type of specialized need, or the impetus may come from one or more of the affected departments. Information Technology departments may be asked to teach such a course because only its faculty members have the necessary technical expertise. Alternately, it may be team taught using one faculty member from Information Technology and another from a client department. An example of this type of course is Programming Fundamentals, given in Appendix B under Integration-First Approach.
- *Single-discipline courses:* These courses are narrower in focus than those discussed in the two preceding sections, and they are generally offered to a homogeneous group of students majoring in a single department. For example, many of us are familiar with a course in discrete mathematics offered by mathematics essentially for Information Technology. This type of course

would fit into the single-discipline category. Another example of this type of course is Integrative Programming, given in Appendix B under Pillars-First Approach.



Computing Curricula Information Technology Volume

Chapter 12 Institutional Challenges

This chapter is a replica of the corresponding chapter in the CS volume of CC2001 with some minor modifications and, as with the CS volume of CC2001, is designed primarily as a resource for colleges and universities seeking to develop or improve four-year programs in Information Technology. To this end, the appendices to this report offer an extensive analysis of the structure and scope of Information Technology knowledge along with a detailed set of course descriptions that represent viable approaches to the four-year curriculum. Implementing a curriculum successfully, however, requires each institution to consider broad strategic and tactical issues that transcend such details. The purpose of this chapter is to enumerate some of these issues and illustrate how addressing those concerns affects curriculum design.

12.1 The need for local adaptation

The task of designing an Information Technology curriculum is a difficult one in part because so much depends on the characteristics of the individual institution. Even if every institution could agree on a common set of knowledge and skills for undergraduate education, there would nonetheless be many additional factors that would influence curriculum design. These factors include the following:

- *The type of institution and the expectations for its degree programs.* Institutions vary enormously in the structure and scope of four-year degree requirements. The number of courses that institutions require of Information Technology majors can vary on the institution type.
- *The range of postgraduate options that students pursue.* Individual schools must ensure that the curriculum they offer gives students the necessary preparation for their eventual academic and career paths.
- *The preparation and background of entering students.* Students at different institutions, and often within a single institution, vary substantially in their level of preparation. As a result, Information Technology departments often need to tailor their introductory offerings so that they meet the needs of their students.
- *The faculty resources available to an institution.* There are limited Information Technology faculty members available to the institutions due to the limited number of graduate programs currently available in the Information Technology area. Therefore, departments need to set priorities for how they will use their limited faculty resources.
- *The interests and expertise of the faculty.* Individual curricula often vary according to the specific interests and knowledge base of the department, particularly at smaller institutions where expertise is concentrated in particular areas.

Creating a workable curriculum requires finding an appropriate balance among these factors, which will require different choices at every institution. There can be no single curriculum that works for all institutions. Every college and university will need to consider the various models proposed in this document and design an implementation that meets the need of that environment.

12.2 Principles for curriculum design

Despite the fact that curriculum design requires significant local adaptation, curriculum designers can draw on several key principles to help in the decision-making process. These principles include the following:

- *The curriculum must reflect the integrity and character of Information Technology as an independent discipline.* Information Technology is a discipline in its own right. That discipline, moreover, is characterized by a combination of theory, practice, knowledge, and skills. Any Information Technology curriculum should therefore ensure that practice is guided both by theory and a spirit of professionalism.
- *The curriculum must respond to rapid technical change and encourage students to do the same.* Information technology is a vibrant and fast-changing field and therefore Information Technology programs must update their curricula on a regular basis. Of equal importance, the curriculum must teach students to respond to change as well. Information Technology graduates must keep up to date with modern developments and should indeed be excited by the prospect of doing so. One of the most important goals of an Information Technology program should be to produce students who are life-long learners.
- *Curriculum design must be guided by the outcomes the program is intended to achieve.* Throughout the process of defining an Information Technology curriculum, it is essential to consider the goals of the program and the specific capabilities students must have at its conclusion. These goals and the associated techniques for determining whether the goals are met provide the foundation for the entire curriculum. In the United States and elsewhere, accreditation bodies have focused increasing attention on the definition of goals and assessment strategies. Programs that seek to defend their effectiveness must be able to demonstrate that their curricula in fact accomplish what they intend.
- *The curriculum as a whole should maintain a consistent ethos that promotes innovation, creativity, and professionalism.* Students respond best when they understand what is expected of them. It is unfair to students to encourage particular modes of behavior in early courses, only to discourage that same behavior in later courses. Throughout the entire curriculum, students should be encouraged to use their initiative and imagination to go beyond the minimal requirements. At the same time, students must be encouraged from the very beginning to maintain a professional and responsible attitude toward their work.
- *The curriculum should be accessible to a wide range of students.* All too often, Information Technology programs attract a homogeneous population that includes relatively few women or students whose ethnic, social, or economic backgrounds are not those of the dominant culture. Although many of the factors that lead to this imbalance lie outside the control of the university, every institution should seek to ensure greater diversity, both by eliminating bias in the curriculum and by actively encouraging a broader group of students to take part.
- *The curriculum must provide students with a capstone experience that gives them a chance to apply their skills and knowledge to solve a challenging problem.* The culmination of a four-year Information Technology degree should include a final-year project that requires students to use a range of practices and techniques in solving a substantial problem. There are aspects of the Information Technology discipline that cannot be presented adequately in the formal classroom setting. These skills can be learned only in the framework of an independent capstone experience.
- *The faculty should constantly be looking for better ways to deliver the curriculum.* Constant improvement in all areas should be a hallmark of a healthy IT program.

12.3 The need for adequate computing resources

Higher education is, of course, always subject to resource limitations of various kinds. At some level, all educational programs must take costs into account and cannot do everything that they might wish to do if they were somehow freed from economic constraints. In many respects, those limitations are no more intense in Information Technology than they are in other academic fields. It is, for example, no longer the case that adequate computing and networking hardware lies outside the reach of academic institutions, as it did in the early days of the discipline. Over the last twenty years, computing and networking equipment have become commodity items, which makes the hardware far more affordable.

At the same time, it is essential for institutions to recognize that computing and networking costs are real. These costs, moreover, are by no means limited to the hardware. Software also represents a substantial fraction of the overall cost of computing and networking, particularly if one includes the development costs of courseware. Providing adequate support staff to maintain the computing and networking facilities represents another large expense. To be successful, Information Technology programs must receive adequate funding to support the computing and networking needs of both faculty and students.

Information Technology is a laboratory discipline with formal, scheduled laboratories included in most of its courses. The laboratory component leads to an increased need for staff to assist in both the development of materials and the teaching of laboratory sections. This development will add to the academic support costs of a high-quality Information Technology program.

12.4 Attracting and retaining faculty

One of the most daunting problems that Information Technology departments face is the problem of attracting faculty. To mitigate the effects of the faculty shortage, we recommend that institutions adopt the following strategies:

- *Adopt an aggressive plan for faculty recruitment.* Scarcity is in itself no reason to abandon the search; the shortage of candidates simply means that Information Technology departments need to look harder. Being successful is usually a matter of initiative and persistence. Departments must start the recruiting process very early and should consider reaching out to a wide range of potential applicants, including overseas students and people currently working in industry.
- *Create academic positions that focus on teaching.* As in most disciplines, faculty positions in Information Technology typically require a Ph.D. and involve expectations in both research and teaching. If there were a sufficient pool of IT candidates with the right credentials and skills, insisting on these qualifications would cause no problem. Given the present shortage of faculty candidates, it is not clear whether Information Technology departments can afford single-minded selectivity. It is not necessary for every institution to maintain a research program in Information Technology. At the same time, it is important for all faculty member to remain current in the field. Because IT is a new discipline, it would be wise to consider candidates from closely-related fields, such as Information Systems and Computer Science. Also, opening faculty positions to those who enjoy teaching but are not drawn to academic research can increase the size of the available pool.
- *Make sure that faculty receive the support they need to stay in academia.* Studies undertaken by the National Science Foundation in the 1980s found that faculty members who left academia for industry typically did not cite economics as their primary motivation [Curtis83]. Instead, they identified a range of concerns about the academic work environment, huge class sizes, heavy teaching loads, inadequate research support, the uncertainty of tenure, and bureaucratic hassles, that the NSF study refers to collectively as "institutional disincentives." As enrollments in Information Technology courses rise, it is critical for institutions to ensure that faculty workloads remain manageable.

- *Get undergraduates involved as course assistants.* Given that there are too few teachers to serve the needs of the many undergraduates, one of the best ways to meet the rising student demand is to get those undergraduates involved in the teaching process. Using undergraduates as course assistants not only helps alleviate the teaching shortfall but also provides a valuable educational experience to the student assistants [[Roberts95](#)].

12.5 Faculty commitment to the degree program

IT programs can make effective use of faculty from a variety of computing disciplines. However, it is essential that there be a core group of dedicated faculty who can provide the right perspective and knowledge to make the program work overall. Specifically, this core group must provide:

- *Experience* – Since IT is a practice-oriented discipline, it is important that many of the faculty have hands-on, practical experience in the core information technologies.
- *Commitment to change* – The rapid evolution of computing requires regular update of all computing programs. For IT, there is a particular need to continue to mirror practice, including continually updating specific technology examples used in labs and demos.
- *Commitment to coordination* – The pervasive themes discussed earlier in this document are central elements in the overall IT degree. Unfortunately, it is difficult to ensure coverage of pervasive themes in courses that have some other topic as a primary focus. For example, coverage of information assurance should be included in programming courses, but coverage may be inadequate or lost completely unless all faculty teaching the courses understand exactly what coverage of pervasive themes is needed. This level of cross course coordination is difficult to achieve and maintain in most institutions of higher education. Without conscious, continual effort by the faculty to communicate and coordinate, coverage of pervasive themes will be uneven at best.

12.6 Conclusion

There is no single formula for success in designing an Information Technology curriculum. Although we believe that the recommendations of this report and the specific strategic suggestions in this chapter will prove useful to a wide variety of institutions, every Information Technology program must adapt those recommendations and strategies to match the characteristics of the particular institution. It is, moreover, important to evaluate and modify curricular programs on a regular basis to keep up with the rapid changes in the field. The Information Technology curricula in place today are the product of many years of experimentation and refinement by Information Technology educators in their own institutions. The curricula of the future will depend just as much on the creativity that follows in the wake of this report to build even better Information Technology programs for undergraduates throughout the world.

Bibliography

- ASPRAY, W., MAYADAS, F., AND VARDI, M. 2006. *Globalization and Offshoring of Software: A Report of the ACM Job Migration Task Force*. Available at <http://www.acm.org/globalizationreport/>.
- BEAULIEU, M. 2002. *Wireless Internet Applications and Architecture*. Addison-Wesley, Boston, MA.
- BOEHM, B.W., AND SULLIVAN, K.J. 2000. Software economics: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (Limerick, Ireland, June 04 - 11, 2000). ICSE '00. ACM, New York, NY, 319-343. DOI= <http://doi.acm.org/10.1145/336512.336584>
- BOHMANN, L.J., MORK, B.A., AND WIITANEN, D.O. 2004. Power engineering design projects: Capstone team projects versus topical design courses. *IEEE Transactions on Power Systems* 19:1, 152-156.
- BUTKUS, M.A., AND KELLEY, M.B. 2004. Approach for integrating professional practice issues into undergraduate environmental engineering design projects. *Journal of Professional Issues in Engineering Education and Practice* 130:3, 166-172.
- CATALANO, G.D. 2004. Senior capstone design and ethics: a bridge to the professional world. *Science and Engineering Ethics* 10:2, 409-415.
- CHUMER, M.J. 2002. Towards An Understanding Of User-Centeredness Within Information Technology Diffusion: A Self-Ethnography. *Doctoral dissertation, Rutgers The State University of New Jersey*. Advisor: Ronald E. Rice.
- CSTB99: Committee on Information Technology Literacy, National Research Council 1999. *Being Fluent with Information Technology*. The National Academies Press, Washington, D.C.
- CURTIS, K.K. 1983. Computer manpower: Is there a crisis? National Science Foundation, Washington D.C. Available at <http://www.acm.org/sigcse/papers/curtis83/>.
- DEVON, R., BILEN, S., MCKAY, A., DE PENNINGTON, A., SERRAFERO, P., AND SIERRA, J.S. 2004. Integrated design: what knowledge is of most worth in engineering design education? *International Journal of Engineering Education* 20:3, 424-432.
- DIAMOND, R. 1998. *Designing and Assessing Courses and Curricula: A Practical Guide, 5th Edition*. Jossey-Bass Publishers, San Francisco, CA.
- DUTTA D., GEISTER, D.E., AND TRYGGVASON, G. 2004. Introducing hands-on experiences in design and manufacturing education. *International Journal of Engineering Education*, 20:5, 754-763.
- EKSTROM, J.J., DARK, M.J., LUNT, B.M., AND REICHGELT, H. 2006. A research agenda for information technology: does research literature already exist?. In *Proceedings of the 7th Conference on Information Technology Education* (Minneapolis, Minnesota, USA, October 19 - 21, 2006). SIGITE '06. ACM, New York, NY, 19-24. DOI= <http://doi.acm.org/10.1145/1168812.1168820>
- EKSTROM, J.J., AND LUNT, B.M. 2003. Education at the seams: preparing students to stitch systems together; curriculum and issues for 4-year IT programs. In *Proceedings of the 4th Conference on Information*

Technology Curriculum (Lafayette, Indiana, USA, October 16 - 18, 2003). CITC4 '03. ACM, New York, NY, 196-200. DOI= <http://doi.acm.org/10.1145/947121.947165>

JONES, S.A., AND HOUGHTALEN R. 2000. Using senior design capstone as model for graduate education. *Journal of Professional Issues in Engineering Education and Practice*, 126:2, 83-88.

MACONACHY, W.V., SCHOU, C.D., RAGSDALE, D., AND WELCH, D. 2001. A model for information assurance: an integrated approach. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, (US Military Academy, West Point, NY, USA, June 5 – 6, 2001).

MELVILLE, N., KRAEMER, K., AND GURBAXANI, V. 2004. Review: information technology and organizational performance: an integrative model of IT business value. *MIS Quarterly*, 28:2.

REDWINE, S.T. (editor) 2007. *Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software*. US Department of Homeland Security. Available at <https://buildsecurityin.us-cert.gov/daisy/bsi/resources/dhs/927.html>.

REICHGELT, H. 2004. Towards a research agenda for information technology. In *Proceedings of the 5th Conference on Information Technology Education* (Salt Lake City, UT, USA, October 28 - 30, 2004). CITC5 '04. ACM, New York, NY, 248-254. DOI= <http://doi.acm.org/10.1145/1029533.1029596>

ROBERTS, E., LILLY, J., AND ROLLINS, B. 1995. Using undergraduates as teaching assistants in introductory programming courses: an update on the Stanford experience. In *Proceedings of the Twenty-Sixth SIGCSE Technical Symposium on Computer Science Education* (Nashville, Tennessee, United States, March 02 - 04, 1995). C.M. WHITE, J.E. MILLER, AND J. GERSTING, Eds. SIGCSE '95. ACM, New York, NY, 48-52. DOI= <http://doi.acm.org/10.1145/199688.199716>

SORK, T., AND CAFFARELLA, R. 1989. Planning Programs for Adults. In *Handbook of Adult and Continuing Education*, S.B. MERRIAM, AND P.M. CUNNINGHAM, Eds. Jossey-Bass Publishers, San Francisco, CA.

The Joint Task Force for Computing Curricula 2005, 2005. *Computing Curricula 2005: The Overview Report*. A cooperative project of ACM, AIS, and IEEE-CS.

Index

Accreditation 2.1, 4.1, 9.1, 9.2, 9.3.3, 9.5, 12.2
ACM Chapter 1 intro, 1.1, 2.1, 9.1, 9.2
Adaptability 4.1, 7.2.1, 9.1, 10.1
Application domains 3.1, 5.2, 8.1.3, chpt 11 intro

Body of knowledge Table of Contents, 1.6, 4.1, 5 intro, 5.1, 5.1.1, 5.1.2, 5.2, 6 intro, 6.3, 6.4, 8.1, 8.2

CC2001 1 intro, 1.1, 1.2, 2 intro, 2.1, 4.1, 5.1.2, 11 intro, 12 intro
CITC 2.1
Communication skills 8.1, 8.1.4, 9.1, 9.3.2, 11 intro, 11.1.3
Complexity 7.1, 7.2.1, 10 intro, 11 intro
Core table of contents, 1.5, 1.6, 4.1, 5.1., 5.1.2, 5.2, 6.1, 6.3, 6.4, chpt 7 intro, 7.1, 7.2, 7.3, 7.4, 7.4.4, 8 intro, 8.2, 9.1, 10 intro

Databases 3.1, 4.1, 5.2, 7.1, 7.3, 7.3.1, 8.2, 10.1
Distance education 3.2

E-commerce 3.1, 8.2
Elective 4.1, 5.1.1, 6.1, 7.4.4, 11.1.1
Ethics 7.2.1, 8.2, 9.1, 9.2, 9.4, 11 intro
Experiential learning 6.2, 6.3

General education 9.2, 11.1.1
Graduate programs 3 intro, 6.3, 12 intro, 12.1

Human-computer interaction 3.1, 4.1, 7.1, 7.3, 7.3.1, 8.2, 11 intro.

IEEE Computer Society
Implementation strategy
Industrial advisory board (IAB) 2.1, 9.3.1, 9.3.3
Internet 3 intro, 3.1, 7.1, 9.4
Interpersonal skills 4.1, 7.2.1, 9.1, 9.3.2, 10.1
IT – Definition 1.3
IT – Broad goals 1.4

IT – Program outcomes 1.5

Knowledge area 4.1, 5 intro, 5.1, 5.2, 6.4, 7.2.1, 7.2.2, 7.3.1, 7.3.2, 7.3.3, 7.4.1, 8.2, 9.3.2, 11 intro

Learning outcomes 4.1, 6.2

Life-long learning 7.2.1, 9.1, 10.1

Mathematics 2.1, 8.1.1, 9.2, 11.1.1, 11.2

Networking 3 intro, 3.1, 4.1, 5 intro, 5.2, 7.1, 7.3, 7.3.1, 8.2, 9.3.2, 10.1, 11 intro

Pervasive themes 4.1, 5.2, 7.2.1, 7.4, 7.4.5, 8.1.7, 10.1

Pillars 6.3, 7.1, 7.3, 7.3.1, 7.4.1

Presentations 6.2, 8.1.4, 9.2, 9.3.1, 10.1, 11.1.3

Probability 8.1.1

Problem solving 1.5, 4.1, 10.1, 11 intro

Professionalism 7.2.1, 9.2, 12.2

Programming 5.2, 7.1, 7.2.2, 7.3, 7.3.1, 7.3.3, 8.2, 9.3.1, 10.1, 11 intro

Responsibility 1.2, 4.1, 5 intro, 7.2.1, 7.4.2, 9.1, 11.1.1

Security 3.1, 4.1, 5 intro, 5.2, 7.1, 7.2.1, 7.4, 7.4.1, 8.2, 9.4, 10.1

SIGITE 1 intro, 1.2, 2.1, 2.2, 5 intro, 5.1.1, 7.1, 7.2.1, 8.1.1, 8.1.2, 8.1.3, 8.1.5, 8.2

SITE 2.1

Statistics 5.2, 8.1.1

Teamwork 5.2, 9.1, 9.2, 9.3.2, 9.4

Tools 3.1, 7.2, 7.2.1, 8.1.1, 8.2, 9.3.1, 9.3.2, 10.1, 11 intro

Web systems 3.1, 5 intro, 5.2, 7.1, 7.3, 7.3.1, 7.3.3, 8.2, 9.1, 11 intro

World-Wide Web (WWW) 3.1, 7.1

Curriculum Committee meetings:

- Full IT Curriculum Committee: July 24, 25, 2003, Calumet, IN, (Purdue-Calumet)
- Full IT Curriculum Committee: after CITC-4, October 2003, West Lafayette, IN (Purdue)
- Writing Subcommittee: early December 2003 in Rochester, NY (RIT)
- Writing Subcommittee: Feb 6,7, 2004 in Statesboro, GA (Georgia Southern)
- Full IT Curriculum Committee, April 2, 3, 2004, Chicago (Purdue-Calumet)
- Writing Subcommittee: June 4,5, 2004, Williamsport, PA (Penn College of Technology)
- Writing Subcommittee: Sept 9-11, 2004, Rochester, NY (RIT)
- Full IT Curriculum Committee: Oct 27, 28, 2004, Salt Lake City, UT (prior to SIGITE 2004)
- Writing Subcommittee: Dec 10, 11, 2004, Philadelphia, PA
- Writing Subcommittee: Feb 11, 12, 2005, Philadelphia, PA
- Full IT Curriculum Committee: April 22, 23, 2005, Rochester, NY (RIT)
- 2-Year Curriculum Committee: April 13, 14, 2007, Rochester, NY (RIT)
- SIGITE Board: April 27, 28, 2007, Atlanta, GA
- Writing Subcommittee: May 28, 29, 2007, Rochester, NY (RIT)
- Steering Committee plus Writing Committee: Feb 8, 9, 2008, Philadelphia, PA



Appendix A

The IT Body of Knowledge

This appendix to the Computing Curricula – Information Technology Volume defines the knowledge domain that is likely to be taught in a four-year curriculum in Information Technology. The underlying rationale for this categorization scheme and additional details about its history, structure, and application are included in the full document. Because we expect the appendices to have wider circulation than the full report, we feel it is important to include in each appendix a summary of the fundamental concepts that are necessary to understand the recommendations. The most important concepts are outlined in the sections that follow.

Structure of the body of knowledge

The IT body of knowledge is organized hierarchically into three levels. The highest level of the hierarchy is the **knowledge area**, which represents a particular disciplinary subfield. Each knowledge area is identified by a two- or three-letter abbreviation, such as or PF for *Programming Fundamentals* or ITF for *IT Fundamentals*. The knowledge areas are broken down into smaller divisions called **units**, which represent individual thematic modules within a knowledge area. Each unit is identified by adding a numeric suffix to the area name; as an example, PF3 is a unit on *object-oriented programming*. Each unit is further subdivided into a set of **topics**, which are the lowest formal level of the hierarchy. In this appendix, there are also some topics which are further broken down into **subtopics**; this has been done simply as an example of what *may* be included in that topic. These are merely examples, and are not intended to be formally prescriptive nor proscriptive.

Core and advanced learning outcomes

We have taken into account the fact that the computing discipline has expanded to such an extent that it is impossible for undergraduates to learn every topic that might be considered fundamental to any particular computing discipline. We have therefore sought to define a minimal **core** consisting of those learning outcomes that are essential to anyone obtaining a four-year degree in this field. Learning outcomes beyond the core outcomes are considered to be **advanced**. By insisting on a broad consensus in the definition of the core outcomes, the task force hopes to keep the core as small as possible, giving institutions the freedom to tailor the advanced components of the curriculum in ways that meet their individual needs.

In creating this curriculum volume, we have found that it helps to emphasize the following points:

- *The core learning outcomes are not a complete curriculum.* Because the core is defined as minimal, it does not, by itself, constitute a complete four-year curriculum. Every four-year program must include additional advanced outcomes from the body of knowledge, although this report does not define what those units will be.
- *Core learning outcomes are not necessarily limited to a set of introductory courses taken early in the four-year curriculum.* Although many of the core outcomes are indeed introductory, there are also some core outcomes that clearly must be covered only after students have developed significant background in the field. For example, we believe that all students must develop a substantial project at some point during their four-year program. The material that is essential to successful management of projects at this scale is therefore part of the core, since it is required of all students. At the same

time, the project course experience is very likely to come toward the end of a student's four-year program. Similarly, introductory courses may include advanced outcomes alongside the coverage of core material. The designation *core* simply means *required* and says nothing about the level of the course in which it appears.

Assessing the time required to cover a unit

To give readers a sense of the time required to cover a particular unit, this report must define a metric that establishes a standard of measurement. Choosing such a metric has proven difficult, because no standard measure is recognized throughout the world. For consistency with the earlier curriculum reports, we have chosen to express time in **hours**, corresponding to the in-class time required to present the material in a traditional lecture-oriented format. To dispel any potential confusion, however, it is important to underscore the following observations about the use of lecture hours as a measure:

- *We do not seek to endorse the lecture format.* Even though we have used a metric with its roots in a classical, lecture-oriented form, we believe that there are other styles -- particularly given recent improvements in educational technology -- that can be at least as effective. For some of these styles, the notion of *hours* may be difficult to apply. Even so, the time specifications should at least serve as a comparative measure, in the sense that a 5-hour unit will presumably take roughly five times as much time to cover as a 1-hour unit, independent of the teaching style.
- *The hours specified do not include time spent outside of class.* The time assigned to a unit does not include the instructor's preparation time or the time students spend outside of class. As a general guideline, the amount of out-of-class work is approximately three times the in-class time. Thus, a unit that is listed as requiring 3 hours will typically entail a total of 12 hours (3 in class and 9 outside).
- *The hours listed for a unit represent a minimum level of coverage.* The time measurements we have assigned for each unit should be interpreted as the *minimum* amount of time necessary to enable a student to perform the learning outcomes for that unit. It is always appropriate to spend more time on a unit than the mandated minimum.

Details of the IT body of knowledge

A summary of the IT body of knowledge -- showing the knowledge areas, units, which units are core, and the minimum time required for each -- appears as Figure A-1. The details of each area follow as separate sections.

The following links will take you to the individual descriptions of these areas:

1. [Information Technology Fundamentals \(ITF\)](#)
2. [Human-Computer Interaction \(HCI\)](#)
3. [Information Assurance and Security \(IAS\)](#)
4. [Information Management \(IM\)](#)
5. [Integrative Programming and Technologies \(IPT\)](#)
6. [Math and Statistics for IT \(MS\)](#)
7. [Networking \(NET\)](#)
8. [Programming Fundamentals \(PF\)](#)
9. [Platform Technologies \(PT\)](#)
10. [System Administration and Maintenance \(SA\)](#)
11. [System Integration and Architecture \(SIA\)](#)
12. [Social and Professional Issues \(SP\)](#)
13. [Web Systems and Technologies \(WS\)](#)

Figure A-1: Summary of IT Body of Knowledge

The Information Technology Body of Knowledge

ITF. Information Technology Fundamentals (25 core hours)

- ITF. Pervasive Themes in IT (17)
- ITF. History of Information Technology (3)
- ITF. IT and Its Related and Informing Disciplines (3)
- ITF. Application Domains (2)

HCI. Human Computer Interaction (20 core hours)

- HCI. Human Factors (6)
- HCI. HCI Aspects of Application Domains (3)
- HCI. Human-Centered Evaluation (3)
- HCI. Developing Effective Interfaces (3)
- HCI. Accessibility (2)
- HCI. Emerging Technologies (2)
- HCI. Human-Centered Computing (1)

IAS. Information Assurance and Security (23 core hours)

- IAS. Fundamental Aspects (3)
- IAS. Security Mechanisms (Countermeasures) (5)
- IAS. Operational Issues (3)
- IAS. Policy (3)
- IAS. Attacks (2)
- IAS. Security Domains (2)
- IAS. Forensics (1)
- IAS. Information States (1)
- IAS. Security Services (1)
- IAS. Threat Analysis Model (1)
- IAS. Vulnerabilities (1)

IM. Information Management (34 core hours)

- IM. IM Concepts and Fundamentals (8)
- IM. Database Query Languages (9)
- IM. Data Organization Architecture (7)
- IM. Data Modeling (6)
- IM. Managing the Database Environment (3)
- IM. Special-Purpose Databases (1)

IPT. Integrative Programming & Technologies (23 core hours)

- IPT. Intersystems Communications (5)
- IPT. Data Mapping and Exchange (4)
- IPT. Integrative Coding (4)
- IPT. Scripting Techniques (4)
- IPT. Software Security Practices (4)
- IPT. Miscellaneous Issues (1)
- IPT. Overview of Programming Languages (1)

MS. Math and Statistics for IT (38 core hours)

- MS. Basic Logic (10)
- MS. Discrete Probability (6)
- MS. Functions, Relations and Sets (6)
- MS. Hypothesis Testing (5)
- MS. Sampling and Descriptive Statistics (5)
- MS. Graphs and Trees (4)
- MS. Application of Math & Statistics to IT (2)

NET. Networking (22 core hours)

- NET. Foundations of Networking (3)
- NET. Routing and Switching (8)
- NET. Physical Layer (6)
- NET. Security (2)
- NET. Network Management (2)
- NET. Application Areas (1)

PF. Programming Fundamentals (38 core hours)

- PF. Fundamental Data Structures (10)
- PF. Fundamental Programming Constructs (10)
- PF. Object-Oriented Programming (9)
- PF. Algorithms and Problem-Solving (6)
- PF. Event-Driven Programming (3)

PT. Platform Technologies (14 core hours)

- PT. Operating Systems (10)
- PT. Architecture and Organization (3)
- PT. Computing Infrastructures (1)
- PT. Enterprise Deployment Software
- PT. Firmware
- PT. Hardware

SA. System Administration and Maintenance (11 core hours)

- SA. Operating Systems (4)
- SA. Applications (3)
- SA. Administrative Activities (2)
- SA. Administrative Domains (2)

SIA. System Integration and Architecture (21 core hours)

- SIA. Requirements (6)
- SIA. Acquisition and Sourcing (4)
- SIA. Integration and Deployment (3)
- SIA. Project Management (3)
- SIA. Testing and Quality Assurance (3)
- SIA. Organizational Context (1)
- SIA. Architecture (1)

SP. Social and Professional Issues (23 core hours)

- SP. Professional Communications (5)
- SP. Teamwork Concepts and Issues (5)
- SP. Social Context of Computing (3)
- SP. Intellectual Property (2)
- SP. Legal Issues in Computing (2)
- SP. Organizational Context (2)
- SP. Professional and Ethical Issues and Responsibilities (2)
- SP. History of Computing (1)
- SP. Privacy and Civil Liberties (1)

WS. Web Systems and Technologies (22 core hours)

- WS. Web Technologies (10)
- WS. Information Architecture (4)
- WS. Digital Media (3)
- WS. Web Development (3)
- WS. Vulnerabilities (2)
- WS. Social Software

Total Hours: 314

Notes:

- Order of Knowledge Areas: Fundamentals first, then ordered alphabetically.
- Order of Units under each Knowledge Area: fundamentals first (if present), then ordered by number of core hours.

IT Fundamentals (ITF) – 25 core hours

ITF. Pervasive Themes in IT

ITF. History of Information Technology

ITF. IT and Its Related and Informing Disciplines

ITF. Application Domains

This knowledge area is intended to be at the introductory level in a curriculum and to provide foundation skills for subsequent courses. It provides an overview of the discipline of IT, describes how it relates to other computing disciplines, and begins to instill an IT mindset. The goal is to help students understand the diverse contexts in which IT is used and the challenges inherent in the diffusion of innovative technology.

ITF. Pervasive Themes in IT

Minimum core coverage time: 17 hours

Topics:

User centeredness and advocacy

Information Assurance and Security

IT systems model

Management of complexity (abstraction, modeling, best practices, patterns, standards, and use of appropriate tools, including versioning and group collaboration tools)

Information and communication technologies

 Human-Computer Interaction

 Information Management

 Networking

 Platform Technologies

 Programming

 Web Systems and Technologies

Adaptability

Professionalism (life-long learning, professional development, ethics, responsibility)

Interpersonal skills

Data versus information

Core learning outcomes:

1. Explain how the components of an IT system interrelate.
2. Explain how and why complexity occurs in IT.
3. Manage complexity in an information technology environment by applying best practices and using appropriate technologies and methodologies.
4. Describe the role of the IT professional as the user advocate.
5. Explain why life-long learning and continued professional development is critical for an IT professional.
6. Explain why adaptability and interpersonal skills are important to an IT professional.
7. Explain the difference between a concept and the possible representations of that concept: for example, the relationship between information and data.
8. Illustrate the use of information and communication technologies to solve problems as an IT professional.
9. Explain why the IAS perspective needs to pervade all aspects of IT.
10. Explain how organizational context is influenced by and impacts the development and deployment of IT systems.

ITF. History of Information Technology

Minimum core coverage time: 3 hours

Topics:

- History of computing technology
- Social history of computing impacts
- Development of user interaction
- History of the Internet

Core learning outcomes:

1. Outline the history of computing technology, the Internet, and the World-Wide Web.
2. Explain how computing and society impact one another.

ITF. IT and Its Related and Informing Disciplines

Minimum core coverage time: 3 hours

Topics:

- Definition of IT
- Computer Science
- Software Engineering
- Information Systems
- Cognitive Science
- Computer Engineering
- Mathematics and Statistics
- Others such as Natural Sciences, Linguistics, Sociology, Psychology, etc.

Core learning outcomes:

1. Explain the relationship between IT and related and informing disciplines.

ITF. Application Domains

Minimum core coverage time: 2 hours

Topics:

- Bio-informatics & medical applications
- Business applications
- Law enforcement
- Political processes
- E-commerce
- Manufacturing
- Education
- Entertainment
- Agriculture
- Software development
- Others

Core learning outcomes:

1. Explain how and to what extent IT has changed various application domains.
2. Explain how IT has impacted the globalization of world economy, culture, political systems, health, security, warfare, etc.

Human Computer Interaction (HCI) – 20 core hours

HCI. Human Factors

HCI. HCI Aspects of Application Domains

HCI. Human-Centered Evaluation

HCI. Developing Effective Interfaces

HCI. Accessibility

HCI. Emerging Technologies

HCI. Human-Centered Computing

A key component to the discipline of Information Technology is the understanding and the advocacy of the user in the development of IT applications and systems. IT graduates must develop a mind-set that recognizes the importance of users and organizational contexts. They must employ user-centered methodologies in the development, evaluation, and deployment of IT applications and systems. This requires graduates to develop knowledge of HCI, including but not limited to such areas as user and task analysis, human factors, ergonomics, accessibility standards, and cognitive psychology.

HCI. Human Factors

Minimum core coverage time: 6 hours

Topics:

Cognitive principles – perception, memory, problem solving, etc.

Understanding the user

Designing for humans – affordances, conceptual models, feedback, constraints, mapping, stages of action, etc.

Ergonomics

Core learning outcomes:

1. Describe the relationship between the cognitive principles and their application to interfaces and products.
2. Explain the conceptual terms for analyzing human interaction with products such as affordance, conceptual model, and feedback.
3. Analyze different user populations with regard to their abilities and characteristics for using both software and hardware products.
4. Explain the importance of user abilities and characteristics in the usability of products.

Advanced learning outcomes:

1. Illustrate how cognitive principles are applied to product design.
2. Design a product for a specific user population.
3. Show how the physical aspects of product design impact its usability.
4. Distinguish between the various principles for designing for humans.
5. Classify the seven stages of action for a task scenario.

HCI. HCI Aspects of Application Domains

Minimum core coverage time: 3 hours

Topics:

Types of environments

Cognitive models

Approaches

Core learning outcomes:

1. Describe different types of interactive environments.
2. Describe the differences in developing user interfaces for different application environments (e.g., Web pages, standalone applications, etc.).
3. Describe several affordances of a Web environment that can enhance the usability of a Web-based application.
4. Explain the connection between the design of a user interface and a model of user domain expertise.
5. Match descriptions of cognitive models with the model names.

Advanced learning outcomes:

1. Develop user interfaces for domain specific applications.
2. Integrate domain experts' knowledge into specific requirements for domain-specific user interfaces.
3. Apply cognitive models to the design of application user interfaces.

HCI. Human-Centered Evaluation

Minimum core coverage time: 3 hours

Topics:

Heuristics
Usability testing
Usability standards – international, OS, accessibility

Core learning outcomes:

1. List the general principles used in the heuristic evaluation of a user interface design.
2. Perform a simple usability evaluation for an existing software application.
3. Classify usability performance and preference metrics: learning, task time, task completion, and user satisfaction.
4. Describe common usability guidelines and standards.

Advanced learning outcomes:

1. Measure the usability of an application by performing a heuristic evaluation.
2. Create an appropriate usability test plan.
3. Measure the usability of a product by analyzing the data from the performance and preference metrics.
4. Recommend changes based upon the results of a usability test.

HCI. Developing Effective Interfaces

Minimum core coverage time: 3 hours

Topics:

Understanding the user experience
Understanding interaction styles
Matching interface elements to user requirements
Graphical user interfaces
Non-graphical user interfaces
Localization, globalization
Development tools - environments, HTML
Prototyping

Core learning outcomes:

1. Explain how the UI and usability affect one another.
2. Define the different types of interaction styles.
3. Select an appropriate UI interaction style for a task.
4. List examples of localization and globalization that would impact design.
5. Implement a simple UI.

Advanced learning outcomes:

1. Describe ways in which users' characteristics (i.e., age, education, cultural differences, etc.) require adaptation of a user interface to increase effectiveness.
2. Describe techniques for developing prototypes of user interfaces.
3. Describe user interface options for devices that do not support visual user interfaces.
4. Critique the differences between low-fidelity and high-fidelity prototyping.
5. Design a low-fidelity GUI prototype for an application which can be evaluated to guide development for a higher-fidelity prototype.
6. Using a prototyping tool, develop a high-fidelity prototype of a GUI which incorporates feedback from early usability testing.
7. Develop prototypes using different I/O modalities to accomplish the same task.
8. Develop prototype interfaces for devices that do not have graphical displays.
9. Given examples of the type of input required from a user, evaluate the effectiveness of different interaction style options.

HCI. Accessibility

Minimum core coverage time: 2 hours

Topics:

Biometrics
Repetitive stress syndrome
Accessibility guidelines and regulations
ADA 508
NIMAS
UDL
WCAG

Core learning outcomes:

1. List some of the advantages and disadvantages of biometric access control.
2. Describe the symptoms of repetitive stress syndrome and list some of the approaches that can ameliorate the problem.
3. Identify sources of accessibility guidelines and standards.
4. List some of the impacts of at least one guideline or standard on designing computer-based applications.
5. Demonstrate the use of accessibility features, such as a narrator.

Advanced learning outcomes:

1. Choose a biometric access system for a given application and include the criteria used for making the choice.
2. Design a new interface to accommodate users with a particular disability.

HCI. Emerging Technologies

Minimum core coverage time: 2 hours

Topics:

- Alternative input/output devices
- Alternative displays (heads-up, goggles, etc.)
- Mobile computing
- Wearable computing
- Virtual reality systems
- Pervasive computing
- Sensor-nets

Core learning outcomes:

1. List several of the emerging alternative I/O devices.
2. Describe the difference between mobile computing and wearable computing.
3. Describe and give examples of pervasive computing.

Advanced learning outcomes:

1. Critique emerging I/O devices as compared to current devices.
2. Modify the design of an application to be used on a mobile device.
3. Describe the ideal characteristics of a wearable display and compare these to what can be done with current technology.
4. List the desirable and undesirable characteristics of a virtual reality system.
5. Illustrate how an emerging device could change the design of an interface for an application.
6. List some characteristics and capabilities of sensor nets.

HCI. Human-Centered Computing

Minimum core coverage time: 1 hour

Topics:

- Human-centered design methods
- Software development lifecycle
- User analysis – profiles, personas
- Social computing
- Task analysis
 - Scenarios
 - Use cases

Core learning outcomes:

1. Explain the characteristics of human-centered design methods.
2. List the advantages and disadvantages for using a human-centered software development approach.
3. Identify a situation in which a user need can be addressed by a software product.
4. Describe, in scenario form, a problem situation to be addressed by a new or redesigned product.

Advanced learning outcomes:

1. Explain how human-computer interaction considerations impact the software development lifecycle.
2. Integrate appropriate user-centered design methodologies into the development of an application or product.
3. Survey a user environment to effectively design software.

4. Analyze a user population and develop generalized profiles for each user group.
5. Compare user groups and develop appropriate personas to represent them.
6. Develop appropriate user personas to be used during the product design phase.
7. Select appropriate user tasks for the application under consideration.
8. Measure the effectiveness of a design of an application or product during different stages of a software development lifecycle.
9. Describe the impact of socialization on the effectiveness of an application interface.

Information Assurance and Security (IAS) – 23 core hours

IAS. Fundamental Aspects

IAS. Security Mechanisms (Countermeasures)

IAS. Operational Issues

IAS. Policy

IAS. Attacks

IAS. Security Domains

IAS. Forensics

IAS. Information States

IAS. Security Services

IAS. Threat Analysis Model

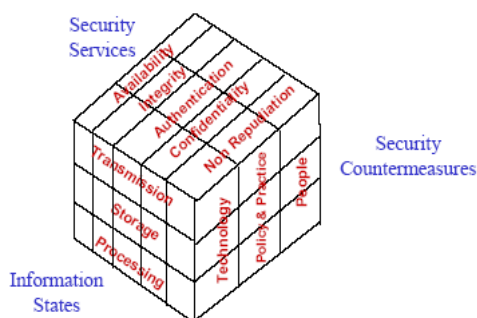
IAS. Vulnerabilities

Since IT systems are increasingly under attack, knowledge of Information Assurance and Security (IAS) is of paramount importance to the profession of IT. The IT professional must understand, apply, and manage information assurance and security in computing, communication, and organizational systems. It is also important for the IT professional to provide users with a framework to be sufficiently security aware to be an asset to the organization rather than a liability. IAS includes operational issues, policies and procedures, attacks and defense mechanisms, risk analyses, recovery, and information security.

It should also be noted that many of the essential educational activities in this knowledge area may be illegal if performed outside a controlled environment, or without proper authorization. It is the responsibility of each individual program to appropriately administer these activities.

There is a model that we have found to be greatly helpful in communicating an IAS model in a relatively small amount of time and in a basic way. This model is shown in the diagram below.

Information Assurance Model



Maconachy, Schou, Ragsdale, Welch: A Model for Information Assurance: An Integrated Approach (bibliography)

IAS. Fundamental Aspects

Minimum core coverage time: 3 hours

Topics:

History and terminology

Security mindset (reasoned paranoia)

Design principles (defense in depth)

System/security life-cycle

Security implementation mechanisms
Gates, guards, guns; cryptography
Information assurance analysis model
MSR model*; threats; vulnerabilities; attacks; countermeasures
Disaster recovery (natural and man-made)
Forensics

Core learning outcomes:

1. Briefly describe the history of the field of *Information Assurance and Security*.
2. Explain the relationship between threats, vulnerabilities, countermeasures, attacks, compromises and remediation.
3. Give examples of how IT system components (e.g. servers, routers, people, software) can be countermeasures, vulnerabilities, and also threats.
4. Explain the security mindset and the role of "paranoia" in that mindset.
5. Explain and give examples of why information assurance and security must be "built in" to design and architecture from the beginning to be most effective.
6. Outline the system life-cycle and its relationship to security.
7. Describe the Security Services as defined by the MSR model.
8. Describe the Information States as defined by the MSR model.
9. Describe the Countermeasures as defined by the MSR model.
10. Given the MSR model, explain how the components interrelate to categorize threats, vulnerabilities and attacks.
11. Describe a disaster recovery scenario.
12. Define forensics.
13. Describe a situation where a forensic investigation would be necessary.

Advanced learning outcomes:

1. Describe the major developments in the field of IAS and analyze the reasons for those developments.
2. Choose a development that has had significant impact on the field of IAS and explain the reason(s) why it is significant.
3. Prepare a threat analysis.
4. Given a threat analysis, recommend the appropriate countermeasures.
5. Demonstrate the ability to preserve evidence necessary for a forensic investigation.

IAS. Security Mechanisms (Countermeasures)

Minimum core coverage time: 5 hours

Topics:

Cryptography
Cryptosystems
Keys: symmetric & asymmetric
Performance (software/hardware)
Implementation
Authentication
"Who you are, what you have, what you know"
Bio-authentication (use of biometrics)
Redundancy
Intrusion detection

Core learning outcomes:

1. Explain the three key factors involved in authentication and how they are used to verify identity and grant access to a system.
2. Explain the process and value of two-factor authentication.
3. Explain the characteristics of an effective password to end-users.
4. Describe and compare physical access control to logical access control.
5. Identify the key types of biometric information utilized in authentication from the perspectives of accuracy, intrusiveness and efficiency.
6. Explain the differences between symmetric and asymmetric cryptosystems, e.g., number of keys required, the types of algorithms used, etc.
7. Explain what is meant by integrity, confidentiality, and authentication.
8. Explain how cryptosystems offer integrity, confidentiality and authentication.
9. Explain digital signatures and certificates.
10. Explain how public key infrastructure (PKI) works.
11. Use a PKI-based application to demonstrate how public-key cryptography works.

Advanced learning outcomes:

1. Describe the single sign-on authentication process and problems related to using and implementing this technology.
2. Compare and contrast key access control and authentication mechanisms to select the one appropriate to specific business contexts (Kerberos, RAS, etc.).
3. Compare the advantages and disadvantages of centralized access controls to decentralized access controls.
4. Describe the DES and 3DES algorithms.
5. Describe the AES algorithm.
6. Explain the differences between block and stream cryptosystems.
7. Explain the differences in efficiency and performance between software based and hardware based cryptosystems.
8. Install and configure a PKI-based application.

IAS. Operational Issues

Minimum core coverage time: 3 hours

Topics:

- Trends
- Auditing
- Cost / benefit analysis
- Asset management
- Standards
- Enforcement
- Legal issues
- Disaster recovery (natural and man-made)

Core learning outcomes:

1. Describe legal and ethical considerations related to the handling and management of enterprise information assets.
2. Specify what constitutes admissible evidence in a legal proceeding and how to acquire and maintain this information.

3. Describe the importance of and key elements involved in incident tracking to develop an incident handling and reporting process.
4. Identify risks associated with disasters or disruptions and specify key mitigation strategies.
5. Identify the types of company assets to be protected by a security plan.
6. Specify the key aspects of physical site security.
7. Describe the elements contributing to the cost of an organization's security management and operations process and their relation to risks and losses associated with information assurance or security related issues and incidents.
8. Describe and evaluate employment policies and practices that are relevant to safeguarding an organization's information assets.
9. Describe the importance of utilizing standards and key standard processes currently utilized in information assurance and their areas of relevance (i.e. DES – Data Encryption Standard).
10. Describe the purpose and elements of the key types of security audits. Discuss how various security standards (i.e. ISO 177799) impact the direction of these audits.

Advanced learning outcomes:

1. Develop an incident handling and reporting process.
2. Recommend appropriate operational and managerial processes to mitigate security and information assurance issues based on a Business Impact Analysis (BIA) report.
3. Create a BIA report that itemizes costs associated with information assurance and security activities and compares these costs to potential risks and costs associated with incidents in these areas.
4. Explain how key information security and assurance standards are or should be utilized in specific industry contexts.
5. Discuss the role of CASPR (Commonly Accepted Security Practices and Recommendations) forms in defining and approving standard operational and management practices.
6. Evaluate how changes in technology and the constantly changing threats impact operational and managerial practices and policies.

IAS. Policy

Minimum core coverage time: 3 hours

Topics:

- Creation of policies
- Maintenance of policies
- Prevention
- Avoidance
- Incident response (forensics)
- Domain integration (physical, network, internet, etc.)

Core learning outcomes:

1. Describe the role of policy and procedure in the IAS Model.
2. Explain why policy and procedure are listed as countermeasures.
3. Explain how poorly defined and executed policies can be a vulnerability.
4. Explain how an organization might develop a policy to defend against password vulnerabilities.
5. Explain why a password policy might need to be modified due to changing circumstances.
6. Explain why security policies must consider all aspects of an organization in order to be effective.
7. Give an example of how vulnerability in one area of an organization might enable a compromise in another area. (Example: weak physical security allows sniffer access to the LAN which allows a password to be read from a POP3 packet. The password is used to gain access to a corporate server.

- Login access to the server allows a root-kit to be applied and the bad guy has total access to the server).
8. Describe a situation in which an incident would require a full forensic approach including evidence gathering, full chain of custody auditing and expert analysis.
 9. Explain how failure to follow good forensic procedures could make prosecution of an attacker impossible.

Advanced learning outcomes:

1. Create a set of policies that implement a specified organizational objective.
2. Justify why a given policy is necessary to meet a specific organizational objective.
3. Update a set of policies to reflect a change in organizational objectives.

IAS. Attacks

Minimum core coverage time: 2 hours

Topics:

Social engineering
Denial of service
Protocol attacks
Active attacks
Passive attacks
Buffer overflow attacks
Malware (viruses, Trojan horses, worms)

Core learning outcomes:

1. Explain and give examples of the social engineering techniques used to gain access to computing and network assets in an organization.
2. Explain how a Denial of Service attack works against an organization's network.
3. List some different protocol attacks to which TCP/IP is susceptible.
4. Explain how the different protocol attacks (e.g. TCP/IP) work against an organization's network.
5. Explain some techniques used during an active attack.
6. Explain some techniques used during a passive attack.
7. Explain how an active attack might use information from a passive attack to compromise a system.
8. Describe and explain how a Buffer Overflow Attack might be used to compromise a system.
9. Identify and distinguish between the different types of Malware (viruses, Trojan horses, worms).

Advanced learning outcomes:

NOTE: Educational activities related to these outcomes should only be carried out in controlled circumstances and with appropriate authorizations.

1. Plan a social engineering attack against an organization's network.
2. Perform a penetration test (active attacks/passive attacks) against an organization's network.
3. Demonstrate the tools and technologies used in passive and active attacks.
4. Demonstrate a Denial of Service attack.

IAS. Security Domains

Minimum core coverage time: 2 hours

Topics:

Security awareness

Possible Domains:

Human-Computer Interaction
Information Management
Integrative Programming
Networking
Program Fundamentals
Platform Technologies
System Administration
System Integration and Architecture
Social and Professional Issues
Web Systems
Physical plant

Core learning outcomes:

1. Give examples of and explain shared concerns across a specified set of security domains.
2. Give examples of and explain concerns that are specific to specified security domains.
3. Discuss the impact of IAS on society and on one's professional and personal practice.

Advanced learning outcomes:

1. Describe the aspects of Human-Computer Interactions which must be considered in designing and implementing secure systems.
2. Explain how Information Management is impacted by security concerns.
3. Explain how Integrative Programming can utilize security and information assurance concepts to ensure the development of secure systems and applications.
4. List and describe several areas of Networking where security must be considered (i.e. access control, authentication, etc.).
5. Discuss several best practices in programming that can prevent security flaws.
6. Explain how security concerns may impact Platform section and operation.
7. Explain key security concerns from a Systems Administration perspective.
8. Discuss policies and practices that may be applied to Systems Integration and Architectures to ensure secure system operation and information assurance.
9. Identify the aspects of Web Systems that must be considered to ensure Web based applications are secure.
10. List the elements of the physical plant that should be considered in preventing security breaches.

IAS. Forensics

Minimum core coverage time: 1 hour

Topics:

Legal systems
Digital forensics and its relationship to other forensic disciplines
Rules of evidence
Search and seizure
Digital evidence
Media analysis

Core learning outcomes:

1. List three types of legal systems used by countries in the world.
2. Describe how digital forensics fits with the other forensic disciplines.

3. Describe a method for capturing a disk drive as evidence and proving its integrity for legal evidence purposes.
4. Explain the difference between the rules for a corporation seizing its property from an employee and law enforcement's seizing of property from a citizen.

Advanced learning outcomes:

1. Explain the concept of "reasonable expectation of privacy", its relationship to corporate policy banners displayed on workstations, and the use of email as evidence in the prosecution of an employee by the corporation.
2. List the areas on a disk that could contain evidence that are not accessible through normal operating system access.
3. List three methods for hiding information on a typical file system.
4. Explain why "free space" often contains interesting data.
5. Capture a forensic image of a disk drive and guarantee the integrity of the image using MD5 or SHA signatures.
6. Perform a basic media analysis of a captured drive using a forensic toolkit.

IAS. Information States

Minimum core coverage time: 1 hour

Topics:

Transmission
Storage
Processing

Core learning outcomes:

1. Give an example of a file representing a document in each of the three states.
2. Justify the statement "An electronic document is frequently in more than one state at the same time."
3. Describe a situation where the same document is in all three states at the same time.
4. Give examples of the relationships between the security services and information states.
5. Give examples of different vulnerabilities that apply to specific states.
6. Explain how the concept of "information state" relates to the concept of a "lifecycle".

Advanced learning outcomes:

1. Analyze a proposed data storage system describing all of the information objects and their states during the entire object lifecycle.
2. Describe some vulnerabilities of the information objects in each of the states.
3. Describe in detail the security services required to implement a specified security policy given an analysis as described above.

IAS. Security Services

Minimum core coverage time: 1 hour

Topics:

Availability
Integrity
Confidentiality
Authentication (source reliability)
Non-repudiation

Core learning outcomes:

1. Describe the possible availability levels for a web service.
2. Explain how redundancy and geographic dispersion relate to availability.
3. Explain the role of integrity, confidentiality, availability, authentication, and non-repudiation as security services.
4. Explain how one-way cryptographic functions are used to implement integrity in document transfer.
5. Explain how cryptographic encryption algorithms are used to implement confidentiality in document transfer.
6. Explain how one-way functions and encryption are used to implement a typical authentication service.
7. Explain how one-way functions are used to implement a non-repudiation service.

Advanced learning outcomes:

1. Describe the sequence of events for a typical Challenge Handshake Authentication Protocol (CHAP) authentication scheme.
2. Describe the sequence of events needed to protect the integrity and ensure non-repudiation of an electronic contract distributed through email to the participants in the contract.
3. Describe what needs to be done in addition to provide confidentiality for the contract in 2 above.

IAS. Threat Analysis Model

Minimum core coverage time: 1 hour

Topics:

- Risk assessment
- Cost benefit

Core learning outcomes:

1. Identify the aspects of a business that may be impacted by a security breach or interruption of operation.
2. Quantify the financial losses associated with potential security breaches and interruption of operations.
3. Identify and describe the nine steps to assess risks associated with security specified by the National Institute of Standards and Technology (NIST).
4. Describe the costs associated with actions that can be taken to mitigate security risks.

Advanced learning outcomes:

1. Perform a comprehensive risk assessment for a specified organization.
2. Create an Information Risk Management (IRM) policy for an organization by analyzing its information assets to determine the appropriate financial, managerial and security aspects of this policy.
3. Identify the quantitative and qualitative measures that can be used to assess risk and evaluate the effectiveness of risk management policies and practices.
4. Specify the value of the benefits to be achieved as a result of risk mitigation efforts and relate this benefit to the cost associated with achieving these benefits.
5. Justify appropriate mitigation strategies by comparing the costs associated with a specific risk and the mitigation strategy.

IAS. Vulnerabilities

Minimum core coverage time: 1 hour

Topics:

- Perpetrators
- Inside attacks
- External attacks
- Black hat
- White hat
- Ignorance
- Carelessness
- Network
- Hardware (design, implementation, installation, etc.)
- Software (design, implementation, installation, etc.)
- Physical access

Core learning outcomes:

1. Define white hat, black hat, hacker and cracker.
2. Explain how culture, community, tools, and technologies contribute to compromising systems.
3. Describe the role of the user in information assurance and how they fit into an overall information assurance plan for an organization.
4. Explain to a non-security community of users what measures they must follow and why, in a situation where their jobs are not security-related.
5. Give an example of how inside and external attacks are similar and are different.
6. List and explain the typical threats and vulnerabilities for an organization's network.

Advanced learning outcomes:

1. Demonstrate how you would test a system for vulnerabilities.
2. Perform a vulnerability analysis of a system.
3. Develop user education modules to educate users on their role in information assurance.
4. Choose a common vulnerability and describe how an attacker uses this vulnerability to gain access to a system.
5. Contrast how various technologies are used by hackers and crackers.
6. Compare and contrast various attack methodologies and differentiate between internal and external attacks.
7. Demonstrate how software contributes to the vulnerabilities of an organization.
8. Explain how design, implementation, and installation of software contribute to the vulnerabilities of an organization.
9. Demonstrate how hardware contributes to the vulnerabilities of an organization.
10. Explain how design, implementation, and installation of hardware contribute to the vulnerabilities of an organization.

Information Management (IM) – 34 core hours

IM. Information Management Concepts and Fundamentals

IM. Database Query Languages

IM. Data Organization Architecture

IM. Data Modeling

IM. Managing the Database Environment

IM. Special Purpose Databases

Information derived from data is important to the management, productivity and differentiation of an organization. Data must be efficiently collected, organized, retrieved and managed to make it meaningful to the organization. It is the role of the IT professional to develop, deploy, manage and integrate data and information systems to support the organization. This knowledge area includes the collection, organization, modeling, transformation, presentation, safety and security of the data and information.

IM. Information Management Concepts and Fundamentals

Minimum core coverage time: 8 hours

Topics:

- Information systems: purpose, use, value
- Properties of data (quality, accuracy, timeliness)
- Database systems
- Analysis of data, forms and sources
- Data collection
- Data retention
- Information backup and recovery

Core learning outcomes:

1. Differentiate and use key terms such as: information, data, database, database management system, metadata, and data mining.
2. Explain the role of data, information, and databases in organizations.
3. Explain how data storage and retrieval has changed over time.
4. Explain the advantages of a database approach compared to traditional file processing.
5. Identify and explain the general types of databases: personal, workgroup, department, enterprise.
6. Explain how the growth of the Internet and demands for information for users outside the organization (customers and suppliers) impact data handling and processing.
7. Define data quality, accuracy and timeliness, and explain how their absence will impact organizations.
8. Describe mechanisms for data collection and their implications (automated data collection, input forms, sources).
9. Explain basic issues of data retention, including the need for retention, physical storage, security.
10. Explain why data backup is important and how organizations use backup and recovery systems.

Advanced learning outcomes:

1. Judge whether given material is information, data or meta-data.
2. Assess the quality, accuracy and timeliness of given data.
3. Determine the data requirements for an application.
4. Determine the data retention requirements for an application.
5. Select the appropriate backup and retention policy for an application and implement it.

IM. Database Query Languages

Minimum core coverage time: 9 hours

Topics:

- SQL data manipulation
- SQL data definition
- SQL performance tuning/optimization
- XQuery and Xpath
- Reports
- Query by example
- Query optimization

Core learning outcomes:

1. Formulate and test SQL queries using SELECT FROM WHERE ORDER BY blocks.
2. Recognize the need for logical operators, set operators, UNION, DISTINCT, LIKE, and BETWEEN operators, and use them appropriately.
3. Formulate and test queries using aggregate functions with GROUP BY HAVING clause.
4. Formulate and test queries using use sub-queries, VIEWS and joins in combinations with the options listed above.
5. Format output (header, footer, totals, subtotals etc.) reports using SQL options and post-processing features of environments like SQL*Plus.
6. Declare appropriate data types, sizes and constraints on elements and their combinations including DATE and TIME types, create TABLE/VIEW with SELECT AS, and use INSERT, UPDATE and DELETE options.
7. Demonstrate an understanding of XPath and XQuery.
8. Formulate and test queries using query by example.
9. Use embedded SQL queries.

Advanced learning outcomes:

1. Use nested correlated sub-queries with (NOT) EXISTS options.
2. Use appropriate substitution variables, SEQUENCES, and OUTER joins.
3. Use advanced (and non-standard) SQL features such tree retrieval using START WITH, and CONNECT BY PRIOR.
4. Improve the performance of SQL queries.
5. Analyze need and use declarative DELETE/INSERT/UPDATE TABLE and ROW level TRIGGERS.
6. Analyze and declaratively define appropriate referential integrity constraints and DELETE/INSERT/UPDATE options SET NULL, SET DEFAULT, CASCADE, and RESTRICT (NO ACTION).
7. Analyze, select and declare physical storage/file structures and options using advanced SQL data definition and related options.
8. Analyze the need for and use report generators and advanced SQL reporting operators.
9. Use statistical optimization and tuning options for evaluating and improving performances of SQL queries including materialized views.

IM. Data Organization Architecture

Minimum core coverage time: 7 hours

Topics:

- Data models

- Hierarchical model
- Network model
- Relational model
- Object databases
- Object-relational databases
- Logical databases
- XML/XMI databases
- Semantic models
- Dimensional models
- Star schema
- Normal forms
 - Functional dependencies
 - 1NF
 - 2NF
 - 3NF
 - BCNF
 - 4NF - multi-valued dependencies
 - 5NF - join dependencies
 - Domain Key NF
 - Second order relations
- Referential integrity
- Entity integrity
- Integrity Constraints
- Indexing

Core learning outcomes:

1. Give a brief history of database models and their evolution.
2. Describe the features of the relational model including relations, tuples, attributes, domains and operators.
3. Demonstrate select, project, union, intersection, set difference, and natural join relational operations using simple example relations provided.
4. List similarities and differences between object-oriented database concepts and features and those of relational databases.
5. Explain the relationship between functional dependencies and keys and give examples.
6. Explain how having normal form relations reduces or eliminates attribute redundancy and update/delete anomalies.
7. Define entity integrity.
8. Define referential integrity.
9. Give examples of user defined integrity constraints.

Advanced learning outcomes:

1. Label 1NF, 2NF, and 3NF violations given a set of relations and a set of functional dependencies.
2. Demonstrate the decompositions necessary to remove normal form violations for specified 2NF and 3NF violations.
3. Given a universal relation R and a set of functional dependencies F , produce a set of relations R such that R contains no BCNF violations or explain why this is not possible for the example.
4. Describe the features added to object-relational systems to distinguish them from standard relational systems.

5. Do back-of-envelope estimates for the time to process a selection or join query, depending on whether an index is or is not present.
6. Choose sensible indices to create, given a set of frequent queries and knowledge of the size of the relations involved.
7. Explain the query plan reported by a DBMS engine.

IM. Data Modeling

Minimum core coverage time: 6 hours

Topics:

- Conceptual Models
 - Entity relationship diagrams
 - Enhanced entity relationship diagrams
 - Identification of business rules
- Logical models
- Physical models
- Reengineering of databases
- Standardized modeling in IDEF1, UML
 - Patterns and standard models
- CASE tools
- Meta-modeling
- Data integration
 - Data warehouses, Data marts

Core learning outcomes:

1. Describe and interpret Entity Relationship diagrams.
2. Create a simple Entity Relationship diagram.
3. Describe and interpret Enhanced Entity Relationship diagrams.
4. Select appropriate business rules for a given scenario.
5. Describe the relationship between a logical model and a physical model.
6. Select a pattern or standard model that effectively corresponds to a given scenario.
7. Explain the use of CASE tools in data modeling.
8. Describe data integration.
9. Describe meta-modeling.
10. Describe a data warehouse, its basic structure, etc.

Advanced learning outcomes:

1. Create and design Entity Relationship diagrams.
2. Create and design Enhanced Entity Relationship diagrams.
3. Formulate and explain identification of business rules.
4. Create and evaluate a logical model.
5. Create and evaluate a physical model.
6. Use a given CASE tool.
7. Demonstrate how to reengineer databases.
8. Apply a pattern or standard model to develop a solution for a given scenario.
9. Create and evaluate meta-models.
10. Explain the concept of data integration and its use in the creation of data warehouses and data marts.
11. Change an existing data warehouse.
12. Change an existing data mart.

IM. Managing the Database Environment

Minimum core coverage time: 3 hours

Topics:

- Data administration
- Database administration
 - Concurrency
 - Security
 - Backup and recovery
- Distributed databases
 - Types (homogeneous, heterogeneous, federated)
- Patterns of distribution (replication, horizontal/vertical partitioning)
- Client-server databases
- n-tier architectures
 - Database connectivity
 - ODBC
 - JDBC
 - XML
 - Web services
 - SOAP

Core learning outcomes:

1. Distinguish between data administration and database administration.
2. Explain the concept of database security.
3. Explain the concept of backup and recovery.
4. Distinguish between homogeneous, heterogeneous and federated distributed databases.
5. Explain the concept of replication as it pertains to distributed databases.
6. Distinguish between horizontal and vertical replication as it pertains to distributed databases.
7. Describe client-server database architecture.
8. Describe n-tier database architecture.
9. Explain the role of ODBC, JDBC and XML in the implementation of an n-tier database architecture.
10. Describe the concept of web services and the role of SOAP.

Advanced learning outcomes:

1. Perform the basic functions of a data administrator, including database planning, analysis, design, implementation, maintenance and protection.
2. Explain different concurrency control protocols.
3. Given a security protocol, secure a database.
4. Given a backup protocol, backup a database.
5. Recover a database.
6. Analyze the appropriateness of a recommendation to replicate (or not replicate) a distributed database in a given situation.
7. Analyze the appropriateness of a partitioning recommendation for a distributed database in a given situation.
8. Use XML in the implementation of an n-tier database architecture.
9. Connect a number of different databases and/or other applications using SOAP.

IM. Special Purpose Databases

Minimum core coverage time: 1 hour

Topics:

- Special purpose databases
 - Text databases
 - Multimedia databases
 - Temporal databases
- Spatial databases
 - Mobile databases
 - Scientific (e.g. genomic) databases
- Decision support
 - Online analytical processing
 - Data warehouses
 - Data mining
- Knowledge management
 - Knowledge (representation and) elucidation
 - Information retrieval
 - Digital libraries

Core learning outcomes:

1. Demonstrate an understanding of online analytical processing and data warehouse systems.

Advanced learning outcomes:

1. Demonstrate an understanding of the role and the concepts involved in special purpose databases such as Full-text, Multimedia, Temporal, Spatial, Mobile, Scientific (e.g. genomic) and other similar database types and effectively use some of them.
2. Declare, integrate and retrieve and analyze data from on-line analytic processing systems and data warehouses using SQL extensions ROLLUP, CUBE, and RANK.
3. Describe methods of data mining and what insights may be gained by these methods.
4. Use data mining visualization and heuristic options in discovery and decision support processes.
5. Demonstrate an understanding of knowledge management.
6. Examine a given Digital Library and comment on how well-structured and/or user friendly it is.

Integrative Programming and Technologies (IPT) – 23 core hours

IPT. Intersystems Communications

IPT. Data Mapping and Exchange

IPT. Integrative Coding

IPT. Scripting Techniques

IPT. Software Security Practices

IPT. Miscellaneous Issues

IPT. Overview of Programming Languages

Organizations typically use many disparate technologies that need to communicate and work with each other. A key component to the discipline of Information Technology is the integration of applications and systems. This knowledge area examines the various types of programming languages and their appropriate use. It also addresses the use of scripting languages, architectures, application programming interfaces and programming practices to facilitate the management, integration and security of the systems that support an organization.

IPT. Intersystem Communications

Minimum core coverage time: 5 hours

Topics:

- Architectures for integrating systems
- DCOM, CORBA, RMI
- Web services and middleware
- Network programming
- Message and queuing services
- Low level data communications

Core learning outcomes:

1. Describe and contrast the different types of architectures for integrating systems.
2. Define the role of DCOM, CORBA, and RMI in distributed processing.
3. Describe how web services are used to integrate disparate applications in an organization: for example, describe the role of the WSDL, SOAP, and UDDI architectures in creating and using web services.
4. Describe the role of socket programming in communicating between systems and contrast the protocols and uses of TCP/IP sockets and Datagram sockets.
5. Describe the purpose of message and queuing services and how they work and list the protocol used by one messaging service (e.g. JMS).
6. List commonly used low level data communications protocols (e.g., RS232), state conditions for when each protocol should be used, and outline the protocol for one low level communications protocol.

Advanced learning outcomes:

1. Create valid WSDL, SOAP and UDDI XML documents to define a web service. Write, debug, and test a web service. Deploy the web service to middleware and invoke the web service from an application across the network.
2. Design, develop and test a socket program that communicates between two different services using both TCP/IP sockets and Datagram sockets.
3. Design, develop and test a program that uses a messaging service to send asynchronous messages to another application across the network.

4. Design, develop and test a program that uses the RS232 protocol to communicate with a hardware device.

IPT. Data Mapping and Exchange

Minimum core coverage time: 4 hours

Topics:

Metadata
Data representation and encoding
XML, DTD, XML schemas
Parsing XML documents
XSL, XSLT and XPath

Core learning outcomes:

1. Define the term metadata.
2. Describe the characteristics of each of the following data encoding schemes, and recommend under what conditions each should be used: ASCII, EBCDIC, and Unicode.
3. Tell how XML and the document object model are being used to integrate and exchange data between systems.
4. Use DTD to create a document definition for a data structure: for example, given a DTD for data structure, create a XML document with real data.
5. Describe how XSL, XSLT and XPath are used to transform data streams.

Advanced learning outcomes:

1. Design, develop and test a program that converts a data stream using one encoding scheme to a different encoding scheme. (EBCDIC and ASCII)
2. Design, develop and test a program that uses SAX or DOM to parse an XML document.
3. Design, develop and test a program that uses XSL and XSLT to transform a data stream from one format to another.

IPT. Integrative Coding

Minimum core coverage time: 4 hours

Topics:

Design patterns
Interfaces
Inheritance

Core learning outcomes:

1. Define the importance of using design patterns and list the motivation for using each of the following design patterns: MVC, singleton, factory method, façade, proxy, decorator, and observer.
2. Describe what a programming interface is and why it is important to programming and give an example of where the use of a programming interface simplified the development of a system.
3. Define the concept of inheritance and describe how it can be applied to encourage code reuse.
4. Design an abstract class and use inheritance to create a class that extends the abstract class.
5. Design, develop and test an application that uses the abstract class.

Advanced learning outcomes:

1. Draw the UML class diagrams for each of the following design patterns: singleton, factory method, façade, proxy, decorator, and observer.
2. Design, develop and test a program that appropriately applies two or more design patterns to solve a problem.
3. Design and develop a programming interface. Implement the interface for at least two different realizations using the factory method pattern. Design, develop and test an application that uses the factory method pattern to instantiate objects for each realization and uses the common interface to access the functionality of each instance.

IPT. Scripting Techniques

Minimum core coverage time: 4 hours

Topics:

Scripting and the role of scripting languages
 Creating and executing scripts
 Influence of scripting on programming

Core learning outcomes:

1. Identify key scripting languages used for web scripting, server-side scripting and operating system scripting.
2. Write, debug and test a script that includes selection, repetition and parameter passing.

Advanced learning outcomes:

1. Write, debug and test a web page that uses scripting to validate the input values in a form.
2. Write, debug and test an interactive web based application that uses server-side script to process input from a web page.
3. Write, debug and test a script using an operating scripting language to facilitate the management of an operating system.

IPT. Software Security Practices

Minimum core coverage time: 4 hours

Topics:

Evidence-based security vs. code access security
 Best security coding practices
 Authentication to system resources and services
 Encryption of data between systems and services

Core learning outcomes:

1. Contrast evidence-based security and code access security.
2. Define the goals of secure coding.
3. Give guidelines for authenticating and defining permissions to systems services and resources.
4. For each of the following “best secure coding” practices, give an example of a problem that can occur when the practice is not followed and then describe how to overcome the problem:
 - a. Preventing buffer overflow
 - b. Securing state data
 - c. Securing method access
 - d. Wrapper code
 - e. Unmanaged code

- f. Validation of user input
- g. Remote considerations
- h. Protected objects
- i. Serialization
- j. Robust error handling

Advanced learning outcomes:

1. Develop and test an application that uses systems services to encrypt a data stream, send it to a different system and decrypt the data stream.
2. Perform a security audit of the code of an existing system, identify problems that violate best security coding practices and make recommendations to fix each problem.

IPT. Overview of Programming Languages

Minimum core coverage time: 1 hour

Topics:

- History of programming languages
- Programming paradigms
- Effects of scale
- Virtual machines
- Compiled vs. interpretative languages
- Application vs. scripting languages

Core learning outcomes:

1. Contrast the differences between the structured and object-oriented programming paradigms.
2. Diagram and label models for both a compiled program and an interpretative program.
3. Describe the benefits and weaknesses associated with using a virtual machine.
4. Give an example where an application language and a scripting language would be more appropriate and give a valid reason to support your selection.

IPT. Miscellaneous Issues

Minimum core coverage time: 1 hour

Topics:

- Adopt & adapt vs. make
- Versioning and version control

Core learning outcomes:

1. List issues that should be considered when deciding whether to create new software or adapt existing software to solve a problem.
2. Tell why it is important to version software and describe one mechanism that can be used to control the versioning of software.

Advanced learning outcomes:

1. Use a version control system to create a new version of a software application, check out a module, make modifications to the module, check the module back in, and then rebuild, deploy and test the new version.
2. Install and set up a version control system for a new project.

Mathematics and Statistics for IT (38 core hours)

MS. Basic Logic

MS. Discrete Probability

MS. Functions, Relations, and Sets

MS. Hypothesis Testing

MS. Sampling and Descriptive Statistics

MS. Graphs and Trees

MS. Application of Mathematics and Statistics to IT

Like any computing discipline, Information Technology relies heavily on concepts from mathematics and statistics. While IT professionals are unlikely to work directly on mathematical or statistical concepts, they will need the ability to manipulate mathematical concepts and to generate and interpret statistical data in order to be successful in their careers. This knowledge unit describes the core concepts and learning outcomes that any graduate from an IT program should have obtained.

It is useful to point out that this knowledge unit only specifies core learning outcomes, i.e. learning outcomes that any graduate from an IT program, independent of their specialization, must have acquired. Depending on their specialization, graduates may need to obtain additional knowledge of mathematics or statistics. For example, students specializing in the areas of networking or platform technologies are likely to need to become comfortable with a range of concepts from calculus, whereas students specializing in the area of information management and/or knowledge management are likely to require a deeper knowledge of statistics than this unit specifies. However, since the model curriculum is agnostic about the type of specializations that institutions create, it seems inappropriate to specify more advanced learning outcomes in detail.

MS. Basic Logic

Minimum core coverage time: 10 hours

Topics:

- Propositional logic
- Logical connectives
- Truth tables and validity
- Predicate logic
- Universal and existential quantification
- Limitations of predicate logic

Core learning outcomes:

1. Apply formal methods of propositional and predicate logic.
2. Create a truth table to determine whether a given formula in predicate logic is valid.
3. Render a well-formed formula in predicate logic in English.
4. Explain the importance and limitations of predicate logic.

MS. Discrete Probability

Minimum core coverage time: 6 hours

Topics:

- Randomness, finite probability space, probability measure, events
- Conditional probability, independence, Bayes' theorem
- Integer random variables, expectation

Probabilistic simulation

Core learning outcomes:

1. Calculate probabilities of events and expectations for random variables.
2. Differentiate between dependent and independent events.
3. Apply the binomial theorem to independent events and Bayes' theorem to dependent events.
4. Apply the tools of probability to create simple discrete event simulations.

MS. Functions, Relations, and Sets

Minimum core coverage time: 6 hours

Topics:

Functions
Relations
Sets and set operations

Core learning outcomes:

1. Explain, with examples, the basic terminology of functions, relations, and sets.
2. Perform the standard operations associated with sets, functions, and relations.
3. Relate practical examples to the appropriate set, functions, or relation model, and interpret the associated operations and terminology in context.

MS. Hypothesis Testing

Minimum core coverage time: 5 hours

Topics:

Formulation of hypotheses: null and alternate hypothesis
Parametric and non-parametric tests and their applicability
Criteria for acceptance of hypotheses, significance levels
t-test, z-test, Chi-square test, and their applicability
Correlation coefficients
Linear and nonlinear regression models
Stochastic versus deterministic analysis

Core learning outcomes:

1. Given a sample situation, formulate an appropriate null hypothesis for a simple question and perform an appropriate test to determine its acceptability.
2. Given a scenario, determine whether a parametric or non-parametric test is appropriate.
3. Discuss the limitations of correlations.
4. Given a sample of two random variables, calculate the t-test, z-test, and Chi-square test statistics and determine if statistical significance exists.
5. Given a set of data, determine the best regression model.
6. Describe the difference between stochastic and deterministic analysis.

MS. Sampling and Descriptive Statistics

Minimum core coverage time: 5 hours

Topics:

Purpose and nature of sampling, its uses and applications

Mean, median, mode
Variance, standard deviation

Core learning outcomes:

1. Explain the purpose and nature of statistical sampling.
2. Distinguish between the concepts of mean, median and mode, and discuss the drawbacks of each as a descriptive statistic.
3. Calculate the mean, median and mode of a given sample of data.
4. Calculate the standard deviation of a given sample of data.

MS. Graphs and Trees

Minimum core coverage time: 4 hours

Topics:

Trees
Undirected graphs
Directed graphs
Spanning trees
Traversal strategies

Core learning outcomes:

1. Illustrate, by example, the basic terminology of graph theory, and some of the properties and special cases of each type of graph.
2. Demonstrate different traversal methods for trees and graphs.
3. Model problems in IT using graphs and trees.

MS. Application of Mathematics and Statistics to IT

Minimum core coverage time: 2 hours

Topics:

Math, statistics and probability in IT

Core learning outcomes:

1. Explain, with examples, the importance of a range of mathematical concepts, including sets, relations, functions, basic logic, and graphs and trees for IT.
2. Explain, with examples, the role of probability and statistics in IT.
3. Perform a statistical analysis of a system's performance.
4. Analyze a statistical analysis of a system's performance and recommend ways to improve performance.

Networking (NET) – 22 core hours

NET. Foundations of Networking

NET. Routing and Switching

NET. Physical Layer

NET. Security

NET. Network Management

NET. Application Areas

Virtually all IT applications involve networking. It is the role of the IT professional to select, design, deploy, integrate, and administer network and communication infrastructures in an organization. This knowledge area includes data communications, telecommunications, inter/intranetworking, and infrastructure security. It also includes application of networking to multimedia, information storage and distribution, and the World Wide Web.

NET. Foundations of Networking

Minimum core coverage time: 3 hours

Topics:

- Standards bodies
- OSI model
- Internet model
- Nodes & links
- LAN, WAN
- Bandwidth, throughput
- Components and architectures
- Routing and switching (bridging)
- Communication protocols

Core learning outcomes:

1. Locate and discuss current standards (i.e. RFC's, IEEE 802 etc) and how standards bodies and the standardization process impact networking technology.
2. Compare and contrast the OSI and Internet models as they apply to contemporary communication protocols.
3. Describe and explain why different technologies are deployed in different contexts of networking, such as topology, bandwidth, distance, and number of users.
4. Explain the basic components and media of network systems and distinguish between LANs and WANs.
5. Explain how bandwidth and latency impact throughput in a data communications channel.
6. Deploy a basic Ethernet LAN and compare it to other network topologies.
7. Configure a client and a server operating system and connect the client machine to the server over a LAN.
8. Analyze and compare the characteristics of various communication protocols and how they support application requirements.
9. Demonstrate the ability to solve basic problems and perform basic troubleshooting operations on LANs and connected devices.

NET. Routing and Switching

Minimum core coverage time: 8 hours

Topics:

- IEEE 802.1
- Routing algorithms
- Routing protocols
- Device architecture
- Advanced routing and switching
- Latency, jitter

Core learning outcomes:

1. Summarize and describe the data communications and telecommunications models, topologies, protocols, standards and architectures in use today.
2. Discuss the concepts and the “building blocks” of today’s data communication networks such as switches, routers, and cabling.
3. Explain the operation and function of 802.1 devices and protocols.
4. Describe the necessary hardware (switches and routers) and components (routing algorithms and protocols) used to establish communication between multiple networks.
5. Analyze the effect of various topologies, applications and devices on network performance topics such as latency, jitter, response time, window size, connection loss and quality of service.

Advanced learning outcomes:

1. Construct multiple networks and connect them together.
2. Analyze and explain routing algorithms and protocols, process routing tables and configure routers for proper operation.
3. Compare and contrast routing protocols and compare/contrast the functions and operation of interior routing protocols with exterior routing protocols.
4. Explain VLANs, trunking and port aggregation and configure and deploy switches utilizing the above.
5. Illustrate how load balancing is accomplished in routers and switches and deploy and test devices utilizing load balancing.
6. Describe strategies to ensure the availability of network access in switched and routed networks.
7. Select appropriate routing and switching equipment for a given network application.

NET. Physical Layer

Minimum core coverage time: 6 hours

Topics:

- Wireless & mobile links
- Switched vs. packets
- Physical media
- Shannon’s law
- Satellite communication
- Error detection and correction
- Communication standards
- Compression & decompression
- Electronics
- IEEE 802

Topologies

Core learning outcomes:

1. Explain how the three variables of Shannon's law impact channel capacity.
2. Critically compare the bandwidth characteristics of several types of physical communication media.
3. Compare and contrast the historical evolution of the switched and routed infrastructures.
4. Analyze the physical challenges inherent in wireless-fixed and wireless-mobile communication channels.
5. Compare and contrast methods of error detection and correction such as parity, CRC, and EDC.
6. Describe how most modern communication standards are developed, addressing both *de jure* and *de facto* standards.
7. Choose the appropriate compression methodology (lossy or lossless) for a given type of application.
8. Analyze and compare four networking topologies in terms of robustness, expandability, and throughput.

Advanced learning outcomes:

1. Compare and contrast the advantages and disadvantages of satellite communication.
2. Calculate the link budget for a given satellite link.
3. Choose a given criterion for rating communication links and order the following from best to worst: wired over unshielded twisted pair; wired over coax; wireless over microwave; wired over optical fiber; wired over shielded twisted pair; wireless over RF (< 1GHz); satellite link.
4. Select appropriate physical media for a given network application.
5. Demonstrate the operation of a phase-locked loop (PLL), and describe how it is used in digital communication.

NET. Security

Minimum core coverage time: 2 hours

Topics:

Fundamentals of cryptography
Applications of cryptography to networks
 Secret-key algorithms; Public-key algorithms; Authentication protocols;
 Digital Signatures; VPN applications
Network attack scenarios
Firewalls
Intrusion detection
Wired, wireless & mobile

Core learning outcomes:

1. Explain how secret-key algorithms protect data confidentiality during transport over a network.
2. Explain how public-key algorithms work to authenticate and how they negotiate secret keys.
3. Explain how SSL protects confidentiality of a TCP connection.
4. Explain how a remote access and a point-to-point VPN work.
5. Explain how firewalls mitigate some network attack scenarios.
6. Explain why weak passwords are a significant problem in networks.
7. Describe a scenario where an intrusion detection system (IDS) could detect a password cracking attempt and what action an intrusion prevention system (IPS) could use to defend against it.

Advanced learning outcomes:

1. Install and configure a firewall.
2. Demonstrate that a firewall is properly configured using a vulnerability testing tool.
3. Install and configure an intrusion detection system (IDS).
4. Demonstrate the function of an IDS by analyzing a simulated intrusion.
5. Demonstrate the function of an IPS by simulating an attack and documenting the results.

NET. Network Management

Minimum core coverage time: 2 hours

Topics:

Management paradigms and protocols
Wireless & mobile
Wired
Security

Core learning outcomes:

1. Discuss typical architectures for network management including the management console, aggregators and device agents.
2. Demonstrate the management of a device such as an enterprise switch through a management console.

Advanced learning outcomes:

1. Compare and contrast various network management techniques as they apply to wired and wireless networks. (This will include topics on devices, users, quality of service, deployment, and configuration of these technologies.)
2. Dissect different security tools and describe implementation strategies of these tools to ensure efficient and effective protection for wired and wireless networks.
3. Describe the configuration and deployment strategies of network devices to optimize network security.
4. Design and implement a local area network.
5. Design a wide area network.
6. Configure the network nodes (computers, routers, etc).
7. Troubleshoot a network problem.
8. Develop and implement a backup and disaster recovery plan as it relates to the network components of an organization.
9. Develop and implement a security plan as it relates to the network components of an organization.
10. Select, install and use appropriate network management tools to collect performance data from a set of network nodes.

NET. Application Areas

Minimum core coverage time: 1 hour

Topics:

Multimedia data technologies
World-wide web
Database and file services

Core learning outcomes:

1. Analyze the impact on the World-wide Web portion of the Internet if the majority of all routers ceased to function.
2. Explain the role of networking in database and file service applications.

Advanced learning outcomes:

1. Install a streaming media server and demonstrate its use from a client application.
2. Explain the implications of multicasting on streaming media delivery.
3. Install and demonstrate a voice communication service such as VOIP.
4. Compare and contrast some of the characteristics of present multimedia data technologies.

Programming Fundamentals (PF) – 38 core hours

PF. Fundamental Data Structures

PF. Fundamental Programming Constructs

PF. Object-Oriented Programming

PF. Algorithms and Problem-Solving

PF. Event-Driven Programming

Programming is a foundational skill for all computing disciplines. This knowledge area develops skills and concepts that are essential to good programming practice and problem solving. It covers fundamental programming concepts, event-driven programming, object-oriented programming, basic data structures, and algorithmic processes. It is important to point out again (see section 7.2.2) that the number of core hours prescribed for this knowledge area is dependent on some previous programming experience.

PF. Fundamental Data Structures

Minimum core coverage time: 10 hours

Topics:

- Primitive types
- Arrays
- Records
- Strings and string processing
- Data representation in memory
- Pointers and references
- Linked structures
- Knowledge of hashing function
- Use of stacks, queues
- Use of graphs and trees
- Strategies for choosing the right data structure

Core learning outcomes:

1. Utilize primitive data types and built-in data structures.
2. Describe common applications for each data structure in the topic list.
3. Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, and queues.
4. Choose the appropriate data structure for modeling a given problem.
5. Describe a simple hash function.

PF. Fundamental Programming Constructs

Minimum core coverage time: 10 hours

Topics:

- Basic syntax and semantics of a higher-level language
- Variables, types, expressions, and assignment
- Conditional and iterative control structures
- Simple I/O
- Functions and parameter passing
- Structured decomposition
- Recursion

Core learning outcomes:

1. Analyze and explain the behavior of simple programs involving the fundamental programming constructs covered by this unit.
2. Modify and expand short programs that use standard conditional and iterative control structures and functions.
3. Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions.
4. Choose appropriate conditional and iteration constructs for a given programming task.
5. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
6. Describe the mechanics of parameter passing and the issues associated with scoping.
7. Describe the concept of recursion and give examples of its use.

Advanced learning outcomes:

1. Implement and trace the execution stack of a simple recursive function.

PF. Object-Oriented Programming

Minimum core coverage time: 9 hours

Topics:

Object-oriented design
Encapsulation and information hiding
Separation of behavior and implementation
Classes and subclasses
Inheritance (overriding, dynamic dispatch)
Polymorphism (subtype polymorphism vs. inheritance)
Class hierarchies
Collection classes and iteration protocols

Core learning outcomes:

1. Discuss and identify the concepts of encapsulation, abstraction, inheritance, and polymorphism.
2. Design, implement, test, and debug simple programs in an object-oriented programming language.
3. Describe how the class mechanism supports encapsulation and information hiding.
4. Design, implement, and test the implementation of “is-a” relationships among objects using a class hierarchy and inheritance.
5. Compare and contrast the notions of overloading and overriding methods in an object-oriented language.
6. Describe the relationship between the static structure of the class and the dynamic structure of the instances of the class.
7. Utilize iterators to access the elements of a container.
8. Describe how constructors and destructors relate to the life of an object.
9. Describe the relationship between an object and its corresponding class.

PF. Algorithms and Problem Solving

Minimum core coverage time: 6 hours

Topics:

- Problem solving strategies
- The role of algorithms in the problem-solving process
- Implementation strategies for algorithms
- Debugging strategies
- The concept and properties of algorithms

Core learning outcomes:

1. Discuss the importance of algorithms in the problem-solving process.
2. Identify the necessary properties of good algorithms.
3. Create algorithms for solving simple problems.
4. Use a programming language to implement, test, and debug algorithms for solving simple problems.
5. Apply effective debugging strategies.

PF. Event-Driven Programming

Minimum core coverage time: 3 hours

Topics:

- Event-handling methods
- Event propagation
- Exception handling

Core learning outcomes:

1. Explain the difference between event-driven programming and command-line programming.
2. Design, code, test, and debug simple event-driven programs that respond to user events.
3. Develop code that responds to exception conditions raised during execution.

Platform Technologies (PT) – 14 core hours

PT. Operating Systems

PT. Architecture and Organization

PT. Computing Infrastructures

PT. Enterprise Deployment Software

PT. Firmware

PT. Hardware

IT professionals will encounter a variety of platforms in their careers. The role of the IT professional is to select, deploy, integrate and administer platforms or components to support the organization's IT infrastructure. This knowledge area includes the fundamentals of hardware and software and how they integrate to form essential components of IT systems.

PT. Operating Systems

Minimum core coverage time: 10 hours

Topics:

- Overview
- Operating system principles
- Concurrency
- Scheduling and dispatch
- Memory management
- Device management
- Security and protection
- File systems
- Real-time and embedded systems
- Fault tolerance
- Scripting
- Virtualization

Core learning outcomes:

1. Describe the necessary components and functions of an operating system.
2. Compare at least two operating systems and evaluate their suitability to a given task or goal.
3. Install a current operating system and validate that the installation was successful.
4. Explain the benefits of using scripts to automate operating systems tasks.
5. Explain concurrency as it relates to scheduling and dispatching.
6. Describe the advantages and issues associated with virtualization.

Advanced learning outcomes:

1. Analyze operating system requirements and recommend an appropriate operating system to meet the requirements.
2. Install several current operating systems and validate that the installations were successful.
3. Write at least one script to perform an operating system task.
4. Install a system with at least two virtual machines.
5. Compare and contrast the memory management strategies of various popular operating systems.
6. Compare and contrast the security models of various popular operating systems.
7. Compare and contrast the file systems of various popular operating systems.
8. Explain the value of fault tolerance for disaster recovery.
9. Explain the implications of virtualization for disaster recovery.

PT. Architecture and Organization

Minimum core coverage time: 3 hours

Topics:

- Machine-level representation of data
- Assembly-level machine organization
- Memory system organization & architecture
- Interfacing and communication
- Functional organization
- Multiprocessing and alternative architectures
- Performance enhancements

Core learning outcomes:

1. Describe how numbers and characters are represented in a computer.
2. Draw a block diagram, including interconnections, of the main parts of a computer.
3. Describe how a computer stores and retrieves information to/from memory and hard drives.
4. Define the terms: bus, handshaking, serial, parallel, data rate.

Advanced learning outcomes:

1. Plan and write a simple assembly-language program.

PT. Computing Infrastructures

Minimum core coverage time: 1 hour

Topics:

- Power and heat budgets
- Servers
- Server farms
- Hardware and software integration

Core learning outcomes:

1. Estimate the power requirements for a computer system.
2. Explain the need for power and heat budgets within an IT environment.
3. Classify and describe the various types of servers and services required within organizations.
4. Describe the need for hardware and software integration.

Advanced learning outcomes:

1. Prepare a computer system for use as a server.
2. Design and implement a multi-computer network and deliver computing services on that network.
3. Perform a cost-benefit analysis for a proposed server solution.
4. Design a server farm for a given situation.
5. Describe three of the major problems associated with current data storage practice.

PT. Enterprise Deployment Software

Minimum core coverage time: 0 hours

Topics:

- Middleware frameworks

Enterprise deployment and management software
Configuration, definition and management
Update management

Core learning outcomes:

None. The coverage time for everyone is 0 hours.

Advanced learning outcomes:

1. Prepare and deploy a system image using imaging software.
2. Identify update management requirements.
3. Create update management plans and policies.
4. Deploy update management software.

PT. Firmware

Minimum core coverage time: 0 hours

Topics:

Firmware storage options
Firmware, software, and hardware
Versioning

Core learning outcomes:

None. The coverage time for everyone is 0 hours.

Advanced learning outcomes:

1. Describe the benefits of firmware storage options.
2. Describe the need for versioning.
3. Distinguish between firmware, software and hardware.

PT. Hardware

Minimum core coverage time: 0 hours

Topics:

Digital logic & digital systems
Benchmarking
Serial vs. parallel
Implementation options
 COTS, PLD, FPGA, gate arrays, custom
Basic electronics
Hardware design languages
Virtual machine emulation

Core learning outcomes:

None. The coverage time for everyone is 0 hours.

Advanced learning outcomes:

1. Design a simple finite state machine with at least 6 states and 4 conditional branches, then build and troubleshoot it.
2. Compare the performance of two different computers with two different operating systems.

3. Rank the five main hardware implementation options in three different scenarios.
4. Compare and contrast the prevalent benchmarks utilized in academia and industry.

System Administration and Maintenance (SA) – 11 core hours

SA. Operating Systems

SA. Applications

SA. Administrative Activities

SA. Administrative Domains

Virtually all organizations have IT needs. It is the role of the IT professional to design, select, apply, deploy and manage computing systems to support the organization. This knowledge area consists of those skills and concepts that are essential to the administration of operating systems, networks, software, file systems, file servers, web systems, database systems, and system documentation, policies, and procedures. This also includes education and support of the users of these systems.

SA. Operating Systems

Minimum core coverage time: 4 hours

Topics:

Installation

Configuration

Maintenance (service packs, patches, etc.)

Server services (print, file, DHCP, DNS, FTP, HTTP, mail, SNMP, telnet)

Client services

Support

Core learning outcomes:

1. Install at least one current operating system.
2. Discuss the importance of system configuration for an organization.
3. Describe the importance of system maintenance for an organization.
4. Identify situations in which a system needs to be reconfigured.
5. Describe when a system requires maintenance.
6. Distinguish between server and client services.
7. Identify situations in which a support organization needs to be consulted in resolving operating system issues.

Advanced learning outcomes:

1. Evaluate various operating systems and recommend a particular operating system to satisfy given needs.
2. Modify the configuration of an operating system.
3. Analyze the pros and cons of installing service packs and updates.
4. Recommend when service packs and operating system updates should be installed.
5. Install service packs and operating system updates.
6. Install various server and client services.

SA. Applications

Minimum core coverage time: 3 hours

Topics:

Installation

Configuration

Maintenance (service packs, patches, etc.)

Server services (database, web, network services, etc.)
Client services
Support

Core learning outcomes:

1. Install at least one current application.
2. Discuss the benefits of custom configuration of applications.
3. Describe the importance of application maintenance for an organization.
4. Identify when an application meets the needs of an organization.
5. Describe when an application no longer meets the needs of an organization.
6. Distinguish between server and client services.
7. Identify situations in which a support organization needs to be consulted in resolving application issues.

Advanced learning outcomes:

1. Summarize several methods to push a custom configuration of applications to users.
2. Assess an application's ability to continue to meet a given organizational need.

SA. Administrative Activities

Minimum core coverage time: 2 hours

Topics:

Content management
Content deployment (file system planning and structure)
Server administration and management
User and group management
Backup management
Security management
Disaster recovery
Resource management
Automation management (automatic job scheduling)
Site management notebooks and documents
System support
User support and education

Core learning outcomes:

1. Describe the need for managing IT resources.
2. Identify situations in which administrative activities are required.
3. Identify situations which interfere with administrative activities.
4. Explain the need for policies governing IT systems.
5. Explain why users need to be trained on IT systems and policies.

Advanced learning outcomes:

1. Explain the benefits of content management within an organization.
2. Explain the need for content deployment.
3. Identify and explain the responsibilities associated with server administration and management.
4. Explain the benefits of managing users and groups.
5. Create policies governing IT systems.
6. Compare and contrast the benefits of automation management.

7. Design and deliver training sessions on IT systems and policies.
8. Compose a timeline for an IT project, given a budget and list of resources.
9. Compare and contrast proactive administrative activities and reactive administrative activities.
10. Prioritize a list of administrative activities for IT, to support an organization's mission statement.

SA. Administrative Domains

Minimum core coverage time: 2 hours

Topics:

- Web domain
- Network domain
- Database domain
- OS domain
- Support domain

Core learning outcomes:

1. Describe the responsibilities common to the various administrative domains.
2. Describe the responsibilities unique to each of the various administrative domains.
3. Identify responsibilities in each domain that support activities in other domains.

Advanced learning outcomes:

1. Justify how you would allocate resources for the various administrative domains.

System Integration and Architecture (SIA) – 21 core hours

SIA. Requirements

SIA. Acquisition and Sourcing

SIA. Integration and Deployment

SIA. Project Management

SIA. Testing and Quality Assurance

SIA. Organizational Context

SIA. Architecture

One of the roles of the IT professional is to design and build systems and integrate them into an organization. This knowledge area develops the skills to gather requirements, then source, evaluate and integrate components into a single system, and finally validate the system. It also covers the fundamentals of project management and the interplay between IT applications and organizational processes.

SIA. Requirements

Minimum core coverage time: 6 hours

Topics:

- Requirements elicitation, documentation, and maintenance
- Modeling requirements
- Use case model
- Modeling tools and methodologies
- Testing
- Project lifecycle phases

Core learning outcomes:

1. Identify the stakeholders of a system and formulate their needs.
2. Compare and contrast the various requirements modeling techniques.
3. Distinguish between non-functional and functional requirements.
4. Identify and classify the roles played by external users of a system.
5. Explain and give examples of use cases.
6. Explain the structure of a detailed use case.
7. Detail a use case based on relating functional requirements.
8. Describe the types of event flows in a use case and under which conditions they occur.
9. Explain how requirements gathering fits into a system development lifecycle.
10. Explain how use cases drive testing throughout the system lifecycle.

Advanced learning outcomes:

1. Select appropriate user tasks for the application under consideration.
2. Summarize the user characteristics for the application under consideration.
3. Assess the appropriate modeling method to use for the proposed information technology in its overall context.
4. Summarize the appropriate deliverables for this phase of the lifecycle.
5. Recommend appropriate changes based upon testing.
6. Create the requirements section of an RFP.

SIA. Acquisition and Sourcing

Minimum core coverage time: 4 hours

Topics:

- Build and buy
- In-sourcing and outsourcing
- System architecture: hardware, software and virtual
- Testing, evaluation and benchmarking
- Contracts and RFPs
- Quality

Core learning outcomes:

1. Differentiate between build and buy in software and hardware acquisition.
2. Explain the advantages and drawbacks of building and buying in general.
3. Differentiate between in-sourcing and out-sourcing for the acquisition of IT services, including support.
4. Explain the advantages and drawbacks of in-sourcing and out-sourcing in general.
5. Explain the importance of testing, evaluation and benchmarking in any IT sourcing decision.
6. Explain the primary components in an RFP.
7. Explain the advantages and drawbacks of using RFPs in an IT sourcing decision.
8. Explain the elements in a well-structured contract.
9. Explain the importance of a well-structured contract in any IT sourcing decision.
10. Given an RFP, recommend and justify one or more products that satisfy the criteria of the RFP.

Advanced learning outcomes:

1. Analyze the appropriateness of a decision to buy or build a software package or a hardware component in a given situation.
2. Analyze the appropriateness of a decision to in-source or out-source IT services in a given situation.
3. Given a set of evaluation criteria, test, evaluate and/or benchmark an IT system, or a component thereof.
4. Create the requirements section of procurement documents including: RFP, RFQ, SOW, etc.
5. Examine a given contract and assess how well-structured it is.

SIA. Integration and Deployment

Minimum core coverage time: 3 hours

Topics:

- Components, interfaces and integration
- Infrastructure, middleware and platforms
- Techniques – data warehouses, extending frameworks, wrappers, glue, facades
- Testing/evaluation/benchmarking
- System release: pilot and acceptance testing and defect repair
- System support strategies and user support plans
- Enterprise integration approaches, standards, and best practices

Core learning outcomes:

1. Define integration in terms of components and interfaces.
2. Give examples of middleware platforms.
3. Discuss some advantages and disadvantages of some middleware platforms.

4. Describe the major considerations for enterprise integration platform selection.
5. Give an example of integration using the “wrapper” approach.
6. Give an example of integration using the “glue code” approach.
7. Give an example of how a framework facilitates integration of components.
8. Explain how the data warehouse concept relates to enterprise information integration.
9. Give examples of how testing and evaluation are impacted by integration choices.

Advanced learning outcomes:

1. Install and test an enterprise integration middleware platform.
2. Develop a component and demonstrate its integration into an existing environment.
3. Identify architecturally significant components in a deployment model.
4. Map a software architecture created in a design to a physical system architecture that executes it.
5. Explain why a pilot deployment is used prior to a system rollout.
6. Identify and explain the tasks involved in a system rollout to all user sites, including consideration of user support issues.
7. Explain how enterprise integration approaches and best practices, such as the Zachman Framework and ITIL, are used by an organization.
8. Explain how standardized (ITIL, COBIT, ISO 20,000) purposes, components, implementation issues, certification, and case studies are used in an organization.

SIA. Project Management

Minimum core coverage time: 3 hours

Topics:

Cost benefit analysis
Roles, responsibilities, accountability
Finance, estimation, budgeting
Planning
Risk management
Scheduling
Tracking
Lessons learned

Core learning outcomes:

1. Explain the key components of a project plan.
2. Explain the importance of a cost/benefit analysis to the successful implementation of a project plan.
3. Explain roles and responsibilities for key project personnel and stakeholders.
4. Use appropriate project planning and tracking tools.
5. Discuss the issues involved in creating a project schedule.
6. Explain how to identify the lessons learned in a project closeout and review session.

Advanced learning outcomes:

1. Working on a team, prepare a project plan for an IT project that demonstrates an ability to follow standard project management methodology.
2. Working on a team, prepare a project plan for an IT project that demonstrates an ability to evaluate project requirements, to define the scope of work, to conduct organizational planning, to identify and evaluate risks, to develop an implementation plan, and to manage change control processes, as well as knowledge of project budgeting, scheduling, and evaluation.
3. Assess how well a project follows its project plan.

SIA. Testing and Quality Assurance

Minimum core coverage time: 3 hours

Topics:

- Standards
- Techniques
- Usability
- Acceptance / contract conformance
- Stress testing
- Performance

Core learning outcomes:

1. Give examples of current testing standards.
2. Explain the various components of usability testing.
3. Explain how an acceptance test is executed and evaluated.

Advanced learning outcomes:

1. Select the appropriate tools and techniques to create a testing environment.
2. Design a stress test that will test and pinpoint the critical issues that might impact system performance.
3. Analyze the results of an acceptance test to determine if the product meets the stated acceptance criteria.
4. Summarize and analyze the data from a usability test and recommend appropriate actions.

SIA. Organizational Context

Minimum core coverage time: 1 hour

Topics:

- Business processes
- IT environment
- Organizational culture

Core learning outcomes:

1. Discuss the relationship between business processes and system integration.
2. Discuss the need to take the current IT environment into account in defining a system architecture and in system integration.
3. Discuss the importance of organizational culture in any system integration project.

Advanced learning outcomes:

1. Explain how business processes influence system architecture.
2. Explain why business processes may have to change as a result of a system integration project.
3. Explain how the current IT infrastructure influences system architecture and system integration.
4. Explain why it is crucial to take organizational culture into account when planning and executing a system integration project.

SIA. Architecture

Minimum core coverage time: 1 hour

Topics:

- Representation/modeling
- Information Architecture
- Enterprise Architecture
- System Architecture
- Enterprise Integration Applications (CRM, ERP)

Core learning outcomes:

1. Explain “architecture” in the context of system integration and architecture. (IEEE Std. 1471)
2. Explain how complex systems can be represented using architectural views and how this facilitates system evolution over time.
3. Explain how some specific architectural views relate to the system lifecycle.
4. Give examples of architectural frameworks and associated best practice models (SOA, Zachman Framework, ITIL, COBIT, ISO 20,000).
5. Give examples of modeling tools that support description and management of architectural views.

Advanced learning outcomes:

1. Describe the major features of a Customer Resource Management system.
2. Describe the major features of an Enterprise Resource Planning system.
3. Explain how Enterprise Architecture relates to Information Architecture and how these impact system architecture.
4. Explain why the concept of views (viewpoints) is so important to managing information technology in the enterprise.
5. Assess how some specific system architecture supports enterprise architecture.
6. Give examples of current tools and methodologies for description and evolution of the various architectural views of an enterprise.

Social and Professional Issues (SP) – 23 core hours

- SP. Professional Communications**
- SP. Teamwork Concepts and Issues**
- SP. Social Context of Computing**
- SP. Intellectual Property**
- SP. Legal Issues in Computing**
- SP. Organizational Context**
- SP. Professional and Ethical Issues & Responsibilities**
- SP. History of Computing**
- SP. Privacy and Civil Liberties**

In addition to technical skills, an IT professional must understand the social and professional context of information technology and computing, and adhere to ethical codes of conduct. This knowledge area covers the historical, social, professional, ethical and legal aspects of computing. It identifies how teamwork is integrated throughout IT and how IT supports an organization. It also stresses professional oral and written communication skills.

SP. Professional Communications

Minimum core coverage time: 5 hours

Topics:

- Oral presentations
- Technical writing
- System documentation
- Technical requirements

Core learning outcomes:

1. Prepare and deliver an oral presentation for a user audience.
2. Prepare and deliver an oral presentation for a management audience.
3. Write a technical memo to management.
4. Create user documentation for an IT system.
5. Create a set of technical requirements for an IT system.
6. Compare and contrast technical writing and expository writing.

Advanced learning outcomes:

1. Create a project portfolio that includes the documents of different phases of a project life cycle.

SP. Teamwork Concepts and Issues

Minimum core coverage time: 5 hours

Topics:

- Collaboration
- Group dynamics
- Leadership styles
- Personality types
- Collaboration tools

Core learning outcomes:

1. Describe personality types and their effect on creating better teams.

2. Describe the basic elements of group dynamics.
3. Compare and contrast different conflict resolution strategies.
4. Compare and contrast basic leadership styles and their effect on teams.
5. Identify and use collaboration tools.
6. Describe ways in which collaboration is used effectively in cross-functional teams.
7. Prepare a self-evaluation of contributions made within a team experience.
8. Prepare a peer evaluation of contributions made by team members.

Advanced learning outcomes:

1. Evaluate collaboration tools and choose appropriate tools for a given project.
2. Choose two published leadership styles. Take a project and evaluate how the usage of each style will affect the project's outcomes in terms of quality of work and results, plus timeliness of project completion.
3. Observe and interview an actual IT group in action near your institution. Report on your findings by analyzing their group dynamics and leadership style.

SP. Social Context of Computing

Minimum core coverage time: 3 hours

Topics:

Social informatics
 Social impact of IT on society
 Online communities & social implications
 Philosophical context
 Diversity issues
 Gender-related issues
 Cultural issues
 Accessibility issues
 Globalization issues
 Economic issues in computing
 Digital divide

Core learning outcomes:

1. Interpret the social context of a particular information technology implementation.
2. Evaluate a particular implementation through the use of empirical data.
3. Describe positive and negative ways in which information technology alters the modes of interaction between people.
4. Explain why computing and networking access is restricted in some countries.
5. Explain the concept of “digital divide”, identify some causes and discuss possible solutions.
6. Identify underlying gender, cultural and diversity related issues in information technology.
7. Identify how information technology changes and affects culture as a whole.
8. Identify how the internet has changed the face of computing and how it has affected society.

Advanced learning outcomes:

1. Compare and contrast the computing and networking access in at least two different countries.
2. Identify some causes of the “digital divide” and discuss possible solutions.

SP. Intellectual Property

Minimum core coverage time: 2 hours

Topics:

- Foundations of intellectual property
- Ownership of information
- Plagiarism
- Software piracy
- Fair use
- Digital Millennium Copyright Act (DMCA)
- Copyrights, patents, trademarks and trade secrets, NDAs
- International differences

Core learning outcomes:

1. Distinguish among copyrights, patents, trademarks and trade secrets.
2. Discuss the ramifications of non-disclosure agreements (NDAs).
3. Discuss the implications of plagiarism, both in education and the profession.
4. Discuss the consequences of software piracy on information technology and the role of relevant enforcement organizations.
5. Discuss how intellectual property laws vary internationally.
6. Describe consequences of the Digital Millennium Copyright Act.

SP. Legal Issues in Computing

Minimum core coverage time: 2 hours

Topics:

- Compliance (ADA508, FERPA, HIPPA, Sarbanes-Oxley, etc.)
- Hackers/crackers
- Computer crime
- Viruses
- System use policies & monitoring
- Risks and liabilities of computer-based systems
- Accountability, responsibility, liability

Core learning outcomes:

1. Identify methods by which computing services can be compromised.
2. Discuss the legal implications of compromising computing services.
3. Discuss the types of policies that should be included for system use and monitoring.
4. Describe the basic elements of compliance laws – such as ADA508, FERPA, HIPPA, and Sarbanes-Oxley.
5. Describe the differences in accountability, responsibility, and liability.
6. Describe current approaches to managing risk, and describe the legal implications of compromising computing services.
7. Evaluate an acceptable use policy.

Advanced learning outcomes:

1. Demonstrate how an application satisfies a particular compliance law.
8. Discuss the consequences of software piracy on software developers and the role of relevant enforcement organizations.

2. Discuss how risk can be managed in an information technology environment.
3. Produce a computer use policy with appropriate enforcement measures.

SP. Organizational Context

Minimum core coverage time: 2 hours

Topics:

Business processes
IT environment
Organizational culture
Professionalism

Core learning outcomes:

1. Outline the basic parts of a typical IT environment.
2. Explain how IT must support business processes.
3. Identify how an IT professional maintains their professional behavior.
4. Explain how an organizational culture can affect IT.

SP. Professional and Ethical Issues & Responsibilities

Minimum core coverage time: 2 hours

Topics:

Relationships with professional societies
Codes of professional conduct, such as IEEE, ACM, BCS, ITAA, AITP
Ethics and history of ethics
Whistle-blowing
Workplace issues (harassment, discrimination)
Identify theft
Ethical hacking

Core learning outcomes:

1. Identify the strengths and weaknesses of relevant professional codes as expressions of professionalism and guides to decision-making.
2. Identify ethical issues that arise in the information technology field and determine how to address them technically and ethically.
3. Apply appropriate professional codes of conduct in assignments.
4. Identify progressive stages in a whistle-blowing incident.
5. List the underlying philosophical aspects of ethical decision making.
6. Identify how information technology is affected by workplace issues such as harassment and discrimination.
7. Identify how society has been affected by identify theft and what to do to protect individuals.
8. Compare and contrast two published codes of ethics.

Advanced learning outcomes:

1. Identify one professional credentialing organization and describe the benefits and drawbacks of professional credentialing.
2. Compare and contrast two published codes of ethics from various areas served by the information technology profession (i.e. healthcare, education, government and business).

SP. History of Computing

Minimum core coverage time: 1 hour

Topics:

Implications of:

History of computer hardware, software

History of the Internet

Telecommunications

The IT profession

IT education

Core learning outcomes:

1. Identify and describe emerging technologies in the context of the history of computing technologies.
2. Identify significant trends in the information technology profession.
3. Identify how life-long learning impacts the information technology professional.

SP. Privacy and Civil Liberties

Minimum core coverage time: 1 hour

Topics:

HIPPA and FERPA

E.U. Data Protection

Gramm-Leach-Bailey Act

Core learning outcomes:

1. List the underlying criteria for both HIPPA and FERPA acts.
2. Identify the basic parts of the E.U. Data Protection act.
3. Discuss the Gramm-Leach-Bailey Act and how it protects privacy.

Web Systems & Technologies (WS) – 22 core hours

WS. Web Technologies

WS. Information Architecture

WS. Digital Media

WS. Web Development

WS. Vulnerabilities

WS. Social Software

IT applications are increasingly web-based. Web technology has grown to include a variety of business, academic, organizational and social applications. Diverse multi-cultural and multi-lingual user communities now depend on Web technology. This knowledge area covers the design, implementation and testing of web-based applications including related software, databases, interfaces and digital media. It also covers social, ethical and security issues arising from the Web and social software.

WS. Web Technologies

Minimum core coverage time: 10 hours

Topics:

- HTTP protocol
- Presentation abstractions
- Web-markup and display languages
- Client-side programming
- Server-side programming
- Web services
- Web servers
- Emerging technologies
- Standards & standards bodies

Core learning outcomes:

1. Describe the structure of the World Wide Web as interconnected hypertext documents.
2. Describe the importance of the HTTP protocol in Web applications.
3. Create and validate HTML/XHTML documents.
4. Demonstrate an understanding of XML syntax and show how to display such documents in Web applications.
5. Use some presentation technologies, such as Cascading Style Sheets and DHTML.
6. Discuss and contrast data entry and validation techniques in client-side vs. server-side programming.
7. Discuss and contrast client-side with server-side security issues.
8. Describe the use of server-side backend databases in web sites and web applications.
9. Describe technologies used in web services, including open source languages and packages, proprietary languages and packages, and enterprise Web development and distributed Web applications.
10. Compare and contrast open source and proprietary server software.
11. Discuss Web standards and standards bodies including the World Wide Web Consortium (W3C).
12. Discuss Web Standards in terms of specifications, guidelines, software, and tools.

Advanced learning outcomes:

1. Apply and integrate HTML/XHTML/XML syntax to create and validate documents.
2. Apply and integrate HTML/XHTML/XML syntax to generate contents via programming.
3. Apply and integrate XML syntax to transform documents between formats.

4. Apply presentation technologies to author websites, such as Cascading Style Sheets, XSLT-FO, DHTML, etc.
5. Evaluate the compliance of web documents with published criteria.
6. Use specialized Web Markup languages, such as SVG, SMIL, RSS, etc.
7. Construct a website to include Client-side programming with JavaScript, Java Applets, Flash, and other Web GUI technologies.
8. Discuss data persistence via cookies in maintaining states.
9. Construct Web server programs with Server-side programming using/working with Generic HTTP CGI techniques, Open Source as well as Proprietary languages and packages.
10. Discuss issues and implementation of server-side security.
11. Discuss issues in maintaining backend databases.
12. Design, construct, test and evaluate web services using Open Source languages and packages, proprietary languages and packages and enterprise web development technology.
13. Design, construct, test and evaluate distributed web applications.
14. Contrast and explain the issues involved in designing web servers including web application servers, streaming media servers, and transformation servers.
15. Assess, compare and select emerging and existing Web technologies for solutions, such as XML, SOAP, WSDL, UDDI, Java, C#, EJB, .NET, etc.
16. Implement Web solutions that comply with Web standards and standard bodies including specifications, guidelines, software, and tools.

WS. Information Architecture

Minimum core coverage time: 4 hours

Topics:

- Hypertext/hypermedia
 - Effective communication
 - Interfaces
 - Navigation schemes
 - Media types
- Web design process
 - User modeling and user-driven design
 - Web design patterns
 - Information organization
 - Usability
 - N-Tier architectures

Core learning outcomes:

1. Build a simple web site that organizes information effectively.
2. Identify alternative ways to organize and present information on a web site.
3. Identify an organization for information based on its inherent structure (chronological, alphabetic, etc.).
4. Identify the purpose of a web site or genre (ecommerce, self-service, educational, governmental service, etc.).
5. Choose a graphic file type that matches the image characteristics and use.
6. Identify time-based media types commonly used on the web.
7. Discuss the use of proprietary media and interaction technologies such as Flash, Active X, RealMedia, and QuickTime.
8. Use cascading style sheets to create style standards for a web site.

9. Describe characteristics of users of a web site that affect design.
10. Describe characteristics that enhance usability of a web site.

Advanced learning outcomes:

1. Identify the purposes of a web site based on client and user interviews.
2. Inventory the content of a web site.
3. Select a site genre that matches the primary purpose of the site.
4. Create a navigational framework that matches the content and genre of the site.
5. Create page templates to maintain style consistency, branding and simplify development of a site.
6. Design an effective homepage for a site.
7. Conduct usability tests for a web site.
8. Explain separation of concerns as it applies to the design and implementation of a web site.
9. Create client-side web-based user interfaces for navigation.
10. Create server-side web-based user interfaces for applications.
11. Implement a three-tier web site.

WS. Digital Media

Minimum core coverage time: 3 hours

Topics:

- Digital libraries
- Media formats
- Capture, authoring and production tools
- Compression
- Streaming media

Core learning outcomes:

1. Describe major digital libraries.
2. Describe media acquisition tools and techniques to create and record media.
3. Describe media acquisition tools and techniques to capture, digitize, and sample media contents.
4. Describe media acquisition tools and techniques for multimedia authoring.
5. Compare/contrast graphic media file format characteristics such as color depth, compression and CODEC.
6. Compare/contrast streaming media file format characteristics such as color depth, bit rate, CODEC and server requirements.
7. Compare/contrast concepts of graphic file formats including lossy/lossless compression, color palettes and CODECs.
8. Discuss the issues involved in deploying/serving media content.

Advanced learning outcomes:

1. Integrate digital libraries with other media contents.
2. Apply media acquisition tools and techniques in creating and recording media.
3. Apply media acquisition tools and techniques in capturing and digitizing media contents.
4. Select and use multimedia authoring tools.
5. Explain and compare media file formats including lossy vs. lossless compression, color palettes, streaming formats, and CODECs.
6. Explain and compare the inter-operability of formats.
7. Deploy and serve media contents within Web applications.

WS. Web Development

Minimum core coverage time: 3 hours

Topics:

- Web interfaces
- Web site implementation and integration
- Database integration
- Accessibility issues
 - Web accessibility initiative

Core learning outcomes:

1. Describe the issues involved in developing a web interface.
2. Summarize the need and issues involved in web site implementation and integration.
3. Explain the importance of interfacing web sites to underlying databases.
4. Explain why accessibility issues are an important consideration in web page development.
5. List some of the organizations that have developed standards for web accessibility and explain the impact that these organizations have had on interface design.

Advanced Learning Outcomes:

1. Design and implement a web interface.
2. Implement a web site and integrate it with other IT applications.
3. Create a web front-end to an underlying database.
4. Design and implement a web site that meets the standards set by such bodies as the Web Accessibility Initiative and/or is compliant with various government mandated regulations, such as section 508 of the US Rehabilitation Act.

WS. Vulnerabilities

Minimum core coverage time: 2 hours

Topics:

- Client Security
 - Cookies and web beacons
 - Phishing
 - Transaction security – certificates and secure connections
 - Spyware
 - Viruses
 - Man-in-the-middle attacks
- Server security
 - Securing a server
 - Securing a host
 - Denial-of-service attacks
- Attacks via clients (e.g. zombies and DOS)
- DNS poisoning

Core learning outcomes:

1. Describe the methods of security for cookies.
2. Identify web sites that are using web page graphics as web beacons.
3. Name ways in which cookies can be used to compromise user privacy.
4. Describe ways to increase the trustworthiness of a website such as security certificates.

5. Describe the use of public key encryption to enhance security.
6. Describe phishing and ways to identify it.
7. Identify ways that spyware is introduced into a user's computer.
8. Describe viruses and how they are introduced into a user's system.
9. Explain what denial of service attacks are and how they are done.
10. Explain how to protect clients from viruses, spyware and zombie processes.

Advanced learning outcomes:

1. Implement client-side and server-side cookies.
2. Set up a web server to support server-side processing in a secure fashion.
3. Identify common server-side configuration issues that affect securing.
4. Implement a security plan to decrease the risks from other users of a shared web server.
5. Design a web application that uses server-side cookies.
6. Describe the mechanism of man-in-the-middle attacks.
7. Explain how DNS Poisoning is used to implement phishing.
8. Use accepted standards to ensure that user input on web pages won't affect server-side processes.
9. Explain how to protect clients from becoming zombies.
10. Clean an infected system of viruses and spyware.

WS. Social Software

Minimum coverage time: 0 hours

Topics:

Asynchronous and synchronous communication modalities
 Broadcast communication modalities
 Collaborative and community modalities
 Ethical issues
 Digital Divide
 Freedom of Speech vs. hate speech
 Pornography
 Privacy
 Copyrights, digital content rights

Core learning outcomes:

None. The coverage time for everyone is 0 hours.

Advanced learning outcomes:

1. Explain the difference between asynchronous and synchronous communication.
2. Summarize and contrast the characteristics of various web-based communication media, such as Listservs, discussion boards, wikis, blogs, and chat-rooms.
3. Describe how the Web has given rise to the emergence of online communities.
4. Debate the various ethical issues associated with the Web, including the Digital Divide, issues concerning race and gender, freedom of speech, privacy, copyright and digital content rights.



Appendix B

IT Course Descriptions

For the purposes of this report, we assume that a **course** meets three times a week over the course of a 15-week semester and that the individual class meetings run somewhere between 50 minutes and an hour. This schedule is typical for a 3-credit semester course in the United States. Given that some of the available time will be taken up with examinations and other activities, we have assumed that 40 hours of lecture are available over the semester. In addition, students are expected to devote three hours of time outside of class for each in-class hour, which means that the total time that each student is expected to invest 160 hours in each course. Other countries use different metrics for expressing the expected level of work. In the United Kingdom, for example, a course described in this report would correspond to 15-16 points under the Credit Accumulation and Transfer Scheme (CATS).

B.1 Curricular approaches

In Chapter 6 [section 6.3](#), two curricular approaches were described: *integration first* and *pillars first*. An example of the integration-first approach is given in detail below, followed by an example of the pillars-first approach.

Integration-First Approach: Introductory Courses

Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
IT Fundamentals	Introduces students to the academic discipline of IT. Pervasive IT themes; IT history; Organizational issues; Relationship of IT to other computing disciplines	None	ITF: Pervasive Themes; History of IT; IT & Related Disciplines; Application Domains. SP: Teamwork Concepts & Issues; Social Context of Computing. MS: Hypothesis Testing.	Pervasive themes in IT; Organizational issues; History of IT; IT and its Related and Informing Disciplines; Application Domains; Applications of Math and Statistics to IT	b; e; g; h; j
Programming Fundamentals	Introduces students to the basics of programming, including data structures, programming constructs, object-oriented programming, algorithms and problem solving, event-driven programming, and recursion.	None	PF: Data Structures; Programming Constructs; Object-Oriented Programming; Algorithms & Problem Solving; Event-Driven Programming. IPT: Intersystems Communications.	Fundamental data structures and programming constructs; Object-oriented programming; Algorithms and problem solving; Event-driven programming; Recursion; Overview of programming languages	a; c; i; j; l; m
Computing Platforms	Principles of computer hardware and low-level software, including logic circuits, assembly language, I/O, storage, program execution; Basics of computer operating systems, including configuration, file systems, security, administration, interfacing, multitasking, performance analysis	IT Fundamentals; Programming Fundamentals	PT: Operating Systems; Architecture & Organization. SA: Operating Systems. SP: Teamwork Concepts & Issues; Social Context of Computing.	Architecture and organization; Computing infrastructures; Operating systems	a; b; c; g; i; j; k; l; m

Integration-First Approach: Intermediate Courses

Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
IT Systems	Introduction to the basic components of IT systems, including networking, web systems, databases, scripting, system administration and maintenance, and system integration	IT Fundamentals; Programming Fundamentals	HCI: Human Factors; Aspects of Application Domains; Evaluation; Developing Effective Interfaces; Accessibility. IM: Concepts & Fundamentals. IPT: Data Mapping & Exchange; Integrative Coding; Scripting Techniques. NET: Foundations; Routing & Switching; Physical Layer; Network Management. PF: Event-Driven Programming. PT: Operating Systems; Architecture & Organization. SA: Operating Systems. SIA: Integration & Deployment. WS: Technologies; Information Architecture; Digital Media; Development.	Foundations of networking; Database systems; Web technologies; Scripting techniques; Integrative coding; Applications; Integration; History of computing	a; b; c; e; g; h; i; j; k; l; m
Web Systems	Introduction to web technologies and systems, including hypertext, self-descriptive text, web page design, web navigational systems, database integration and digital media	IT Systems	IAS: Fundamental Aspects; Security Mechanisms; Attacks; Security Domains; Forensics; Information States; Security Services; Threat Analysis Model. IM: Database Query Languages; Data Organization Architecture; Data Modeling. IPT: Data Mapping & Exchange; Integrative Coding; Scripting Techniques. WS: Technologies; Information Architecture; Digital Media; Development; Vulnerabilities.	Data administration; Database administration; Scripting techniques; Integrative coding; Web technologies; Information architecture; Digital media; Web development; Vulnerabilities	a; b; c; d; e; f; g; i; j; k; l; m
Networking	Builds a deeper understanding of how networks work, including the topics of LANs, WANs, service providers, packets, hubs, routers, switches, Internet protocols	IT Systems	NET: Foundations; Routing & Switching; Physical Layer; Security.	Routing and switching; Physical layer; Security; Application Areas	a; b; e; g; i; j; l; m
Databases	Builds a deeper understanding of how databases work, including the topics of database theory and architecture, data modeling, normalization, query languages, security, and Web applications	IT Systems	IM: Concepts & Fundamentals; Database Query Languages; Data Organization Architecture; Data Modeling; Managing Database Environment; Special Purpose Databases.	Data organization and retrieval; Data modeling; Database query languages; Database systems; Information models and systems; Properties of data; Specification of data requirements; Data and database administration	a; b; c; d; e; f; g; i; j; k; l; m

Human-Computer Interaction	Introduction to the basic concepts of human-computer interaction, including human factors, performance analysis, cognitive processing, usability studies, environment, and training	IT Systems	HCI: 1, 2, 3, 4, 5, 6, 7	Human factors; HCI aspects of application domains; Human-centered evaluation; Developing effective interfaces; Emerging technologies; Human-centered software; Accessibility	b; c; d; e; f; g; i; j; k; l; m
Technical and Professional Communications	Introduction to written and oral technical and professional communication, including proposals, reports, presentations, formal papers	College English	SP: Communications	Professional Communications	f; h

Integration-First Approach: Advanced Courses

Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
Information Assurance & Security	Introduction to the concepts of data security, including policies, attacks, vulnerabilities, encryption, information states, and forensics	Networking; Databases; Web Systems	IAS: Fundamental Aspects; Security Mechanisms; Operational Issues; Policy; Attacks; Security Domains; Forensics; Information States; Security Services; Threat Analysis Model; Vulnerabilities.	Security mechanisms; Fundamental aspects; Security services; Information states; Threat analysis model; Vulnerabilities; Attacks; Policy; Operational issues; Forensics; Security domains	b; e; g; i; j; l; m;
IT Capstone I	IT senior project-first semester, including project proposal, feasibility studies, intellectual property, teamwork, budgets, schedule management	Networking; Databases; Human-Computer Interaction	SIA: Requirements; Acquisition & Sourcing; Integration & Deployment; Project Management; Organizational Context; Architecture. SP: Communications; Teamwork Concepts & Issues; Intellectual Property; Legal Issues in Computing; Organizational Context; Professional & Ethical Issues & Responsibilities; History of Computing; Privacy & Civil Liberties.	Requirements; Acquisition/sourcing; Integration; Project management; Architecture; Teamwork concepts and issues; Intellectual property; Organizational context	b; d; e; f; g; h; i; j; l; m; n
IT Capstone II	IT senior project-second semester, including teamwork, professional communications (reports and presentations), design implementation, testing	IT Capstone I	SA: Applications; Activities; Domains. SIA: Integration & Deployment; Project Management; Testing & Quality Assurance. SP: Communications; Teamwork Concepts & Issues; Intellectual Property.	Testing and QA; Organizational context; Professional communications; Teamwork concepts and issues;	b; d; e; f; g; h; i; j; l; m; n
Professional Ethics	Covers all the areas of ethics in the computing profession	Networking; Databases; Human-Computer Interaction	SP: History of Computing	Professional and Ethical Issues and Responsibilities; Privacy and Civil Liberties	b; e; f; g

Pillars-First Approach¹: Introductory Courses

This would likely match best with 2-year programs or 2+2 programs.

Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
IT Fundamentals	Introduces students to the academic discipline of IT as well as the general meaning of IT.	None	ITF: all SP: (3)	ITF: Pervasive Themes in IT ITF: History of IT ITF: Related & Informing Disciplines ITF: Application Domains MS: Applications of Math & Stats to IT SIA: Organizational Context SP: Social Context of Computing SP: Organizational Context SP: History of computing	b; f; g; h; j
Programming Fundamentals	Introduces students to the basics of programming. Includes data structures, programming constructs, algorithms and problem-solving, object-oriented and event-driven programming, and recursion.	None	PF: all	PF: Fundamentals of Data Structures PF: Fundamentals Programming Constructs PF: Object-Oriented Programming PF: Algorithms and Problem-Solving PF: Event-Driven Programming	c; f; g; j
Fundamentals of Networking	Introduces students to the fundamentals of networks and networking in IT. Includes routing, switching, physical layer, security, and application areas.	None	NET: all	NET: Foundations of Networking NET: Routing and Switching NET: Physical Layer NET: Security NET: Network Management NET: Application Areas	a; i; j
Fundamentals of Web Systems	Introduces students to web systems and technologies. Includes information architecture, digital media, web development and vulnerabilities of web systems.	Programming Fundamentals	WS: all core	WS: Web Technologies WS: Information Architecture WS: Digital Media WS: Web Development WS: Vulnerabilities	a; b; c; g; j; l; m
Technical and Professional Communication	Introduces teamwork concepts, group dynamics, leadership styles, technical writing and documentation, presentation development and delivery.	College English	SP: (2)	SP: Professional Communications SP: Teamwork Concepts and Issues	d; f

¹ This sample largely addresses only the core outcomes of the IT curriculum. Additional coursework would be needed to flesh out a complete degree.

Pillars-First Approach: Intermediate Courses

Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
Fundamentals of Information Management	Introduces students to databases and information management. Includes query languages, data organization architecture, data modeling, managing the database environment and special-purpose databases.	Programming Fundamentals	IM: all	IM: IM Concepts and Fundamentals IM: Database Query Languages IM: Data Organization Architecture IM: Data Modeling IM: Managing Database Environment IM: Special-Purpose Databases	a; b; c; e; g; j; l
Fundamentals of Human Computer Interaction	Introduces students to HCI. Includes human factors, HCI aspects of application domains, human-centered evaluation, developing effective interfaces, accessibility, emerging technologies, human –centered software development	Programming Fundamentals; or Fundamentals of Web Systems (dependent on how faculty wants to handle HCI)	HCI: all	HCI: Human Factors HCI: HCI Aspects of Application Domains HCI: Human-Centered Evaluation HCI: Developing Effective Interfaces HCI: Accessibility HCI: Emerging Technologies HCI: Human-Centered Computing	b; c; e; g; i; j; k; l; m
System Administration and Maintenance	Introduces students to system administration and maintenance as well as platform technologies. Includes operating systems, applications, administrative activities and domains, computer architecture and organization, and computing infrastructures.	Programming Fundamentals	PT: all core SA: all	PT: Operating Systems PT: Architecture and Organization PT: Computing Infrastructures SA: Operating Systems SA: Applications SA: Administrative Activities SA: Administrative Domains	e; g; j; k; l
Integrative Programming	Introduces students to integrative programming. Includes intersystems communication, data mapping and exchange, integrative coding, scripting techniques, software security and an overview of programming languages.	IT Systems; Fundamentals of Information Management	IPT: all	IPT: Intersystems communications IPT: Data Mapping and Exchange IPT: Integrative Coding IPT: Scripting Techniques IPT: Software Security Practices IPT: Miscellaneous Issues IPT: Overview of Programming Languages	a; c; j; k; l; m
Information Assurance and Security	Introduces students to IAS. Includes fundamental aspects, security mechanism, operational issues, policy, attacks, security domains, forensics, information states, security services, threat analysis and vulnerabilities.	Programming Fundamentals; Fundamentals of Web Systems; Fundamentals of Information Management	IAS: all	IAS: Fundamental Aspects IAS: Security Mechanisms IAS: Operational Issues IAS: Policy IAS: Attacks IAS: Security Domains IAS: Forensics IAS: Information States IAS: Security Services IAS: Threat Analysis Model IAS: Vulnerabilities	b; j; m

Pillars-First Approach: Advanced Courses

Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
System Integration and Architecture	Includes requirements analysis, acquisition, sourcing, integration, project management, testing and quality assurance, organizational context and architecture.	Integrative Programming	SIA: all	SIA: Requirements SIA: Acquisition and Sourcing SIA: Integration and Deployment SIA: Project Management SIA: Testing and Quality Assurance SIA: Organizational Context SIA: Architecture	a; b; d; e; g; j; k; l; m; n
Information Assurance and Security II	Provides a follow-up to IT380. Includes advanced outcomes from IAS.	Information Assurance and Security	IAS: advanced outcomes	NA	b; j
IT Capstone	IT senior project including project proposal, feasibility studies, intellectual property, teamwork, budgets, schedule management; professional communications (reports and presentations), design implementation, testing.		SIA: all SP: (5)	SIA: Requirements SIA: Acquisition and Sourcing SIA: Integration and Deployment SIA: Project Management SIA: Testing and Quality Assurance SIA: Organizational Context SIA: Architecture SP: Professional Communications SP: Teamwork Concepts and Issues SP: Social Context of Computing SP: Intellectual Property SP: Organizational Context	a; b; d; e; f; g; j; k; l; m; n
Social and Professional Issues in IT	Includes professional communications, history of computing, social and organizational context of computing, Teamwork concepts and issues, intellectual property, legal issues, professional and ethical issues, privacy and civil liberties.	Integrative Programming ; Information Assurance and Security	SP: all	SP: Professional Communications SP: Teamwork Concepts and Issues SP: Social Context of Computing SP: Intellectual Property SP Legal Issues in Computing SP: Organizational Context SP: Professional and Ethical Issues and Responsibilities SP: History of Computing SP: Privacy and Civil Liberties	d; e; f; g; j

Practicum in the Curriculum

The placement of practicum in the curriculum is best decided by each program individually. Practicum courses and/or internships may be used to supplement the integration-first approach or the pillars-first approach discussed in this document. To prepare students for practicum opportunities, the early courses of either approach would have substantial experiential learning incorporated into them. This presupposes that before students can engage in an internship, they must have had sufficient depth in one or more IT areas to give them an adequate skill set.



Appendix C

Learning Outcomes/Knowledge Areas Matrix

The IT Model Curriculum was designed in such a way that successful implementation increases the likelihood that a program will be accredited. The ABET CAC accreditation criteria require that the curriculum is designed in such a way that all the program outcomes for a specific program can be achieved. This appendix is one way to show how the model curriculum meets this criterion. The left column lists the program outcomes that are specified in the ABET CAC IT program-specific accreditation criteria. The right column gives the knowledge areas or knowledge units that address each learning outcome. The entry "PerTheme" refers to the pervasive themes that are addressed multiple times throughout the IT curriculum and do not really belong to any single knowledge area or unit. These pervasive themes are discussed previously in 7.2.1.

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline	ITF; HCI. Developing Effective Interfaces; IM; IPT; MS; NET; PF; PT; WS; PerTheme4
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution	HCI.Human Factors, Developing Effective Interfaces, Accessibility; IAS.Operational Issues, Threat Analysis Model; IM.Concepts & Fundamentals; IPT.Software Security Practices; NET.Physical Layer, Application Areas; PT.Operating Systems, Computing Infrastructures; SA.Operating Systems, Applications; SIA.Requirements, Acquisition & Sourcing; WS.Information Architecture; PerTheme1,2,4,7
(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs	HCI.Human Factors, Aspects of Application Domains, Evaluation, Developing Effective Interfaces, Accessibility; IM.Concepts & Fundamentals; IPT.Data Mapping & Exchange, Integrative Coding, Software Security Practices; NET.Network Management; PF.Programming Constructs, Object-Oriented Programming, Event-Driven Programming; PT.Computing Infrastructures, Enterprise Deployment Software; SA.Operating Systems, Applications, Activities; SIA.Integration & Deployment, Testing & Quality Assurance; SP.Teamwork Concepts & Issues; WS.Technologies, Digital Media, Development, Vulnerabilities; PerTheme1,4,7
(d) An ability to function effectively on	SIA. Project Management; SP. Teamwork Concepts & Issues, PerTheme7

teams to accomplish a common goal	
(e) An understanding of professional, ethical, legal, security and social issues and responsibilities	ITF. Pervasive Themes; IAS. Operational Issues, Security Domains, Forensics, Vulnerabilities; SP. History of Computing, Privacy & Civil Liberties; WS. Social Software; PerTheme6
(f) An ability to communicate effectively with a range of audiences	ITF. Pervasive Themes; SP. Communications, Organizational Context; PerTheme7
(g) An ability to analyze the local and global impact of computing on individuals, organizations, and society	ITF; HCI. Developing Effective Interfaces; IAS. Security Domains; SP. Social Context of Computing, Legal Issues in Computing, Organizational Context, Professional & Ethical Issues & Responsibilities, History of Computing, Privacy & Civil Liberties; WS. Social Software; PerTheme1
(h) Recognition of the need for and an ability to engage in continuing professional development	ITF; PerTheme5,6
(i) An ability to use current techniques, skills, and tools necessary for computing practice	ITF; HCI. Human Factors, Aspects of Application Domains, Evaluation, Developing Effective Interfaces, Software; IAS; IM; IPT; NET; PF; PT; SP. History of Computing; WS. Technologies, Information Architecture, Digital Media, Development, Vulnerabilities; PerTheme2,4,5
(j) An ability to use and apply current technical concepts and practices in the core information technologies	ITF; HCI. Human Factors, Aspects of Application Domains, Evaluation, Developing Effective Interfaces, Software; IAS; IM; IPT; NET; PF; PT; SP. History of Computing; WS. Technologies, Digital Media, Development, Vulnerabilities; PerTheme1,2,4
(k) An ability to identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems	ITF1; HCI. Human Factors, Software; SA. Operating Systems, Applications, Activities; SIA. Requirements, Organizational Context; PerTheme1,4
(l) An ability to effectively integrate IT-based solutions into the user environment	HCI. Developing Effective Interfaces; IM. Data Modeling; IPT. Data Mapping & Exchange, Integrative Coding, Scripting Techniques; NET. Network Management; PT. Computing Infrastructures; SA; SIA. Acquisition & Sourcing, Integration & Deployment, Testing & Quality Assurance, Organizational Context; WS. Information Architecture; PerTheme2
(m) An understanding of best practices and standards and their application	ITF. Pervasive Themes; HCI. Evaluation; IAS. Security Mechanisms, Operational Issues, Threat Analysis Model; IPT; NET. Foundations, Routing & Switching, Physical Layer; SIA. Architecture; SP. Professional & Ethical Issues & Responsibilities; WS. Technologies, Digital Media; PerTheme2,3
(n) An ability to assist in the creation of an	ITF. History of IT, IT & Related Disciplines; SIA. Project Management; PerTheme3

effective project plan	
------------------------	--