Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

Acknowledgements

Members of the Select 2-Year Committee - Writing Subcommittee: (in alphabetical order)

Deborah Boisvert	University of Massachusetts Boston	Boston, MA
Robert Deadman	Ivy Tech Community College	Lafayette, IN
Sandra Gorka	Pennsylvania College of Technology	Williamsport, PA
Jacob Miller	Pennsylvania College of Technology	Williamsport, PA
Paula Velluto	Bunker Hill Community College	Charlestown, MA

Members of the Select 2-Year Committee: (in alphabetical order)

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

Michael Bilynsky	Corning Community College	Corning, NY
Diane Delisio	Miami University	Miami, OH
Eugenia Fernandez	Indiana U - Purdue U Indianapolis	Indianapolis, IN
Sandra Gorka	Pennsylvania College of Technology	Williamsport, PA
Priscilla Grocer	Bristol Community College	Fall River, MA
James Hightower	California State Fullerton	Fullerton, CA
Eydie Lawson	Rochester Institute of Technology	Rochester, NY
Barry Lunt	Brigham Young University	Provo, UT
Joseph Mani	Modern College of Bus. and Science	Sultanate of Oman
Julie Mariga	Purdue University	West Lafayette, IN
Dawn Miller	Roxbury Community College	Boston, MA
Keith Morneau	Capella University	Minneapolis, MN
Rob Murray	Ivy Tech Community College	Terre Haute, IN
Joan Osborne	Lord Fairfax Community College	Warrenton, VA
Tim Preuss	MN State Community and Technical College	Moorhead, MN
Maria Rynn	Northern Virginia Community College	Annandale, VA
Ashraf Saad	Armstrong Atlantic University	Savannah, GA
Mark Stockman	University of Cincinnati	Cincinnati, OH
Paula Worthington	Northern Virginia Community College	Woodbridge, VA

Additional Proofing of Final Document:

Helen Yoas	Pennsylvania College of Technology	Williamsport, PA

Editors and Co-Chairs of Above Committees:

Deborah Boisvert	University of Massachusetts Boston	Boston, MA
Paula Worthington	Northern Virginia Community College	Woodbridge, VA

SIGITE Vice-Chair for Education

Barry Lunt	Brigham Young University	Provo, UT

Special Interest Group for IT Education (SIGITE)

Two-Year Information Technology Guidelines

Chapter 0
Two-Year College Environment

According to the American Association of Community Colleges, approximately one-half of all undergraduates in the United States are enrolled in community colleges, and more than half of all first-time college freshmen attend community colleges. "Community colleges are centers of educational opportunity. They are an American invention that put publicly funded higher education at close-to-home facilities, beginning nearly 100 years ago with Joliet Junior College [in Joliet, Illinois]. Since then, they have been inclusive institutions that welcome all who desire to learn, regardless of wealth, heritage, or previous academic experience. The process of making higher education available to the maximum number of people continues to evolve" (http://www.aacc.nche.edu/).

The two-year college environment is uniquely positioned, resulting from the threefold mission of these institutions to provide a learning environment for:

- transfer into baccalaureate programs;
- entrance into the local workforce; and
- lifelong learning for personal and professional enrichment.

In addition, many two-year colleges are drivers of local economic development, providing workforce development and skills training, as well as offering noncredit programs ranging from English as a second language to skills retraining to community enrichment programs or cultural activities.

Two-year colleges serve high school graduates proceeding directly into college, workers needing to upgrade skill sets or master new ones in order to re-enter the workforce, immigrants seeking to become integrated into the local culture and master a new language, individuals leaving the workplace to engage college-level coursework for the first time, returning students with college degrees who have decided to pursue an alternate career path, and many individuals in need of ongoing training and skill updating. This diversity is addressed in numerous ways, including targeted career counseling, remediation of basic skills, specialized course offerings, individualized instruction and attention, flexible scheduling and delivery methodologies, and a strong emphasis on retention and successful completion. Furthermore, because two-year colleges have less restrictive entrance requirements, faculty must be prepared to instruct students exhibiting a broad range of academic preparations, aptitudes, and learning styles. The mission of two-year college faculty is to focus their full-time attention on effective pedagogy for educating a diverse student population, remaining current in their discipline and in the scholarship of teaching and learning, and fostering student success.

Two-year, community, and technical colleges award associate degrees to students completing two years of study, as do certain four-year colleges. These associate-degree programs are complete in their own right, whether designed specifically to enable graduates to transfer into the upper division of a baccalaureate program or to allow them to gain entry into the workforce. These institutions also offer certificate programs, intended to be fulfilled in less time than a complete degree program; such programs are often designed for targeted student audiences and focused on specific content.

At the earliest opportunity, faculty and academic advisors must help each student determine which type of program best serves the student's educational and career goals. Such considerations include the distinctions between certificate, career, and transfer programs, the academic requirements of each, and the associated employment options. Career-oriented associate degree programs provide the specific knowledge, skills, and abilities necessary to proceed directly into the workplace, while transfer-oriented programs provide the academic foundation and pathway to continue a program of study at a four-year college or university.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

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. This Subcommittee was tasked with 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*. This process was completed in November 2008 with the publication of the IT Volume at the ACM Website.

Since fall 2003, the SIGITE Curriculum Committee has intended to complete a similar document to provide guidance for associate degree programs. In spring 2007, a SIGITE Select 2-Year Committee was formed and met to review the then state of the IT Volume and its the outcomes to determine what could be accomplished at institutions offering 2-year degrees. In October 2008, this group established a Select 2-Year Writing Subcommittee to undertake the task of creating a first draft of an IT Volume for 2-year degree programs that could be submitted to the (CCECC) of the ACM.

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, and Software Engineering. A draft of *CC 2005: The Overview Report* was used in the early development of 4-year Information Technology Volume (IT2008); the final version of *CC 2005: The Overview Report* has also been used in completing IT2008.

1.2 Overview of the process for developing this IT volume

The decision was made to complete this volume following the acceptance of the 4-year IT Volume. In general, the information in this volume will lag behind the content in the 4-year IT Volume. As such, this document should be updated immediately after any updates to the IT Volume. Additionally, when using this volume one should also obtain the 4-year IT Volume to determine if this volume is in need of updating. We, the authors of this document, acknowledge the process used in developing the structure and content of the CC2001 document and the IT2008 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*.

The content of the knowledge areas in this document is largely a subset of the knowledge areas in the IT Volume. There are, however, some instances in which the Committee decided to expand on the learning outcomes that are presented in the IT Volume.

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 associate degree program

The principle goal of an A.S. program is to provide graduates with the skills necessary to transfer to a baccalaureate program in information technology in order to complete their education. While not all of the program outcomes of a 4-year program are relevant for the associate's degree, it is important to have the trajectory in perspective. Specifically, students completing a bachelor's IT program of study should be able to demonstrate:

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.

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 two-year programs in IT.

This document is intended to describe a <u>typical</u> curriculum for an IT program designed for transfer to another 4-year degree. It is fully anticipated that there will be many variations and flavors of IT programs, including those that would lead to employment after the completion of the two-year degree.

Because of the broad, integrative nature of IT, the core includes basic coverage of a high proportion of all the units in the body of knowledge. 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 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 two-year report, as we have attempted to build on the strengths of the four-year IT Volume and other Computing Curricula while avoiding some of the problems observed in earlier reports or volumes. Chapters 5 and 6 present overviews of the two-year 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 the first two years of an IT program. Because these courses reflect a subset of the four-year program which is not in itself a complete 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 content and pedagogical approaches 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 two-year Information Technology programs. Appendix B consists of full descriptions for the recommended courses that comprise a sample approach. 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.

Special Interest Group for IT Education (SIGITE)

Two-Year Information Technology Guidelines

Chapter 2
The Context of this Report

In developing this report, the SIGITE Select 2-Year Committee did not have to start from scratch. We have benefited tremendously from the work that resulted in the *Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology* that was formally accepted by ACM in December 2008. That document took into account the previous work contained in the *Computing Curricula 2001* report and *Computing Curricula 2005: The Overview Report*. This group also reviewed materials relevant to the two-year college educator that included *Guidelines for Associate Degree Programs to Support Computing in a Networked Environment – 2000 (Information Technology)* and *Guidelines for Associate-Degree Transfer Curriculum in Computer Science – 2009* produced by the ACM Two-Year College Education Committee (TYCEC). Additionally, many of the members of this committee benefited from participating in a National Science Foundation-funded CPATH project: *A Community addressing Seamless Information Technology Education for Students (CSITES)* – (CCF#0722237). 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 discipline began in the Fall of 2001 with informal meetings between faculty in IT programs at a small number of institutions. A curriculum committee was formed and began working on the IT volume in December of 2003, using as a model the CS volume of the *CC2001* document. A draft of *CC2005: The Overview Report* became available in 2005 and was also used. In October 2005, a draft of the 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 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.

The Special Interest Group for Information Technology Education (SIGITE) of the ACM has been pivotal in formulating accreditation criteria for programs in computing in general and IT in particular. SIGITE recognizes that the boundaries that define IT are somewhat fluid, and will continue to change. However, experience has shown that there is a durable core body of knowledge that characterizes IT. This 2-Year document, like the 4-Year Volume, consists of this core component that every 2-year college will teach to all IT students. It also provides an advanced component comprised of elective topics from which 2-year colleges can select to offer specialized courses to its students. In explicitly recognizing the importance of the two-year college in the information technology computing arena, SIGITE has provided generous opportunities for community college input at its conferences. The result has been this model curriculum for 2-year IT programs. One compelling feature of this work is the ability to provide for seamless transition from 2-year to 4-year institutions. This document is being regarded as a comprehensive approach to facilitate a seamless transfer to a 4-year program (2+2). By utilizing the model curriculum to assess seamless education from 2 to 4 year institutions we envision a high degree of continuity, transferability and articulation.

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.

Special Interest Group for IT Education (SIGITE)

Two-Year Information Technology Guidelines

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. In addition, the past several decades have seen extraordinary increases in workplace productivity due primarily to the impact of technology, and specifically Information Technology (IT). During this timeframe, IT has become the essential enabler of global communication and commerce. Virtually every enterprise (regardless of size) and every industry sector is completely dependent on the flow and exchange of data and information. Increasingly, employers want students to integrate business, technology, and industry specific skills that can be applied to another field, such as in healthcare, energy, biotechnology, and manufacturing, just to name a few. There is a significant and consistent demand for employees who select, create, apply, integrate and administer computing technologies in ways that further the work of the enterprise; persons who adapt quickly to rapid technological advances and who can thereby maintain the enterprise's competitive advantage. Globalization has dispersed IT tools, skills and knowledge throughout the world. While basic computing and IT skills are now among the world's most available commodities, innovation (critical to long term success) amplifies the continual need for sophisticated and innovative computing knowledge and methodologies. Individuals who have the ability to manage and transform data and information to produce value for the enterprise are in great demand. This creates a corresponding need for people with the capabilities to conduct data-driven research for product development and process improvement, and for medical and scientific research: the long term drivers of economic growth. The rapid evolution of the discipline has a profound effect on Information Technology education, affecting both content and pedagogy.

3.1.1 Changes in Content:

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

- The World Wide Web and internetworking
- Networking technologies
- Systems administration and maintenance
- Graphics and multimedia
- Web systems and technologies
- Service-oriented architecture
- E-commerce technologies
- Relational databases
- Storage
- Cloud computing
- Virtualization
- Data analytics
- Client-server technologies
- Interoperability and convergence
- Technology integration and deployment
- Object-oriented event-driven programming
- Application programmer interfaces (APIs)
- Human-computer interaction
- Security and forensics
- 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 Changes in Pedagogy

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, simulations, and other interactive media, combined with computer projection and individual laboratory stations, have made a significant difference in the way Information Technology can and should be taught. The design of Information Technology curricula must take those changing technologies into account, with focus in the areas of computational thinking, scenario development, active/cooperative learning strategies, problem-based learning, communications, technology, leadership, and teamwork.

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 that 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."

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. It is also important that outcomes from both 2- and 4-year programs provide upward mobility and add value at each level. 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's skill set 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 3-1 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.

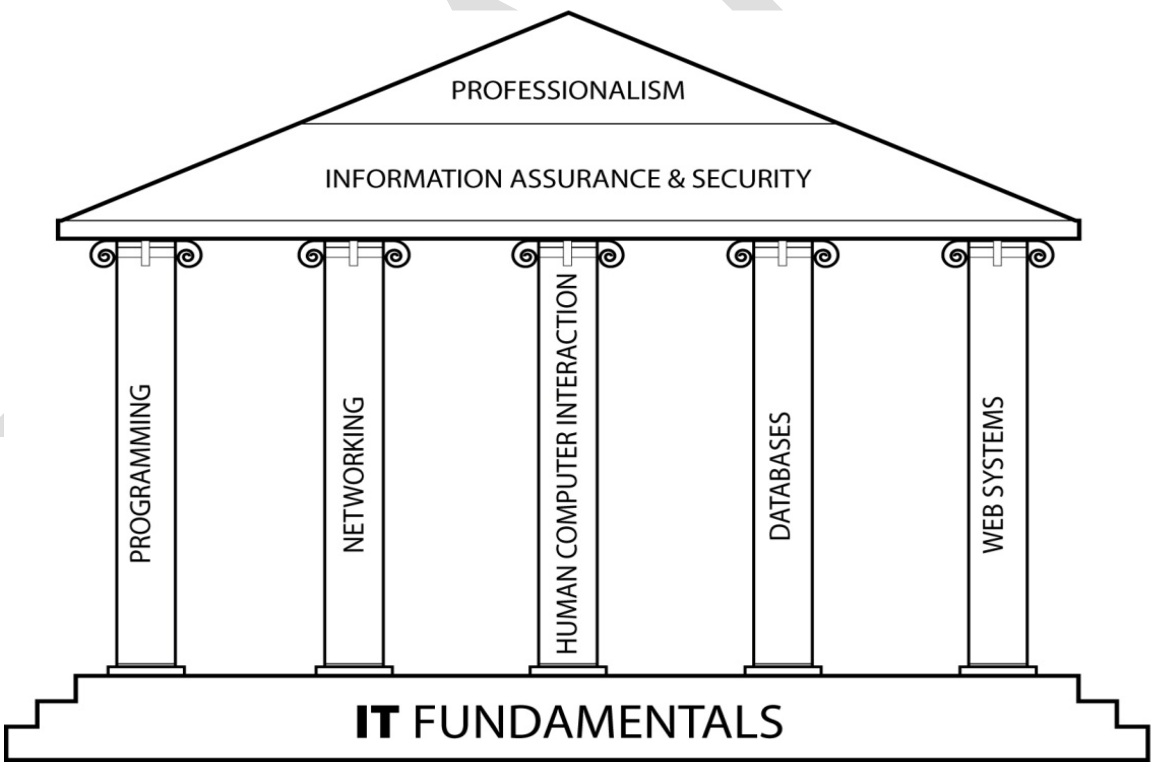

Figure 3-1. The Information Technology Discipline

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

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 for articulation and transfer to 4-year institutions*. This 2-Year document, like the 4-Year Volume, consists of a core body of knowledge that every 2-year college will teach to all IT students. It also provides an advanced component comprised of elective topics from which 2-year colleges can select to offer specialized courses to its students. This document is being regarded as a comprehensive approach to facilitate a seamless transfer to a 4-year program (2+2).

2. *This document is intended to exist as a companion to the IT Volume 2008 contained within the ACM Computing Curriculum 2005 series. This document followed the format of Information Technology 2008: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology 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. *This document is intended to be sensitive to the special considerations unique to the two-year college environment, including the need to:*

- serve under-prepared students;
- limit course proliferation and be efficient in class offerings;
- offer students opportunities both for transfer into baccalaureate programs and for direct entry into the workplace;
- augment the technical curriculum with general education courses;
- ensure that students have the interpersonal and communication skills necessary to succeed in today's global society;
- support student exploration of various educational and career choices;
- serve the needs of local business and industry for a trained workforce; and
- provide lifelong learning for personal and professional enrichment. (http://acmtyc.org/curricula.cfm)

4. *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.

5. *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.

6. *The curriculum must reflect both the unique nature of Information Technology as well as 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. 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.

7. *This document is aimed at two-year transfer 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 within and across two-year programs and other types of educational initiatives, we expect aspects of this document to be broadly applicable.

8. *The development of this volume has been 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. By utilizing the body of knowledge to define outcomes and provide seamless education from 2-year to 4-year institutions, we envision a high degree of continuity, transferability and articulation. Two-year colleges can now focus on an agreed upon and educationally validated core body of knowledge, which will serve to mitigate course-by-course equivalence or, what is worse, constant renegotiation of transfer agreements for matters as trivial as textbook or staff changes. Once again, it is important to note that this document applies predominantly to transfer programs, but is useful in informing and creating a common language that can be used across a broader set of computing pathways.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

Chapter 5
Overview of the IT Body of Knowledge

In developing a curriculum for two-year study in Information Technology, one of the first steps is to identify the four-year approved curriculum and organize the material that would be appropriate for the two-year level in a similar format. As noted in Chapter 1, we sought to accomplish this goal by convening a set of knowledge area focus groups comprised of two-year and 4-year representatives and addressing each one the learning outcomes the body of knowledge associated with 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 leaning outcomes are skills that anyone obtaining a two-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 at any level. A baccalaureate 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 could be a blueprint for a four-year transfer degree.* Because the core is defined as minimal, it does not, by itself, constitute a complete two-year curriculum.
- *Core learning outcomes are not necessarily achieved in a set of introductory courses early in the two-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 basic system, including hardware and software installation and testing, at some point during their two-year program. The material that is essential to successful management of projects at this level is therefore part of the core, since it is required of all students. At the same time, the project course experience should come at the end of a student's two-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.

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 an example 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 Two-Year Body of Knowledge

ITF. Information Technology Fundamentals (25 hours)
ITF1. Pervasive Themes in IT (17)
ITF2. History of Information Technology (3)
ITF3. IT and Its Related and Informing Disciplines (3)
ITF4. Application Domains (2)

HCI. Human Computer Interaction (11 hours)
HCI1. Human Factors (2)
HCI2. HCI Aspects of Application Domains (2)
HCI3. Human-Centered Evaluation (2)
HCI4. Accessibility (2)
HCI5. Emerging Technologies (2)
HCI6. Human-Centered Software Development (1)

IAS. Information Assurance and Security (13 hours)
IAS1. Fundamental Aspects (2)
IAS2. Security Mechanisms (Countermeasures) (3)
IAS3. Operational Issues (2)
IAS4. Policy (2)
IAS5. Attacks (2)
IAS6. Forensics (0.5)
IAS7. Threat Analysis Model (0.5)
IAS8. Vulnerabilities (1)

IM. Information Management (25 hours)
IM1. IM Concepts and Fundamentals (8)
IM2. Database Query Languages (8)
IM3. Data Organization Architecture (7)
IM4. Data Modeling (2)

IPT. Integrative Programming & Technologies (10 hours)
IPT1. Intersystems Communications (1)
IPT2. Data Mapping and Exchange (2)
IPT3. Scripting Techniques (4)
IPT4. Software Security Practices (1)
IPT5. Overview of Programming Languages (1)
IPT6. Miscellaneous Issues (1)

MS. Math and Statistics for IT (17 hours)
MS1. Functions, Relations and Sets (6)
MS2. Basic Logic (10)
MS3. Application of Math & Statistics to IT (1)

NET. Networking (14 hours)
NET1. Foundations of Networking (3)
NET2. Routing and Switching (7)
NET3. Physical Layer (2)
NET4. Security (1)
NET5. Network Management (0)
NET6. Application Areas (1)

PF. Programming Fundamentals (35 hours)
PF1. Fundamental Data 7)
PF2. Fundamental Programming Constructs (10)
PF3. Object-Oriented Programming (9)
PF4. Algorithms and Problem-Solving (6)
PF5. Event-Driven Programming (3)

PT. Platform Technologies (12 hours)
PT1. Operating Systems (8)
PT2. Architecture and Organization (3)
PT3. Computing Infrastructures (1)

SA. System Administration and Maintenance (6 hours)
SA1. Operating Systems (3)
SA2. Applications (1)
SA3. Administrative Activities (2)

SIA. System Integration and Architecture (8.5 hours)
SIA1. Requirements (4)
SIA2. Integration and Deployment (1)
SIA3. Project Management (1)
SIA4. Testing and Quality Assurance (2)
SIA5. Organizational Context (0.5)

SP. Social and Professional Issues (16.5 hours)
SP1. Professional Communications (2)
SP2. Teamwork Concepts and Issues (5)
SP3. Social Context of Computing (2)
SP4. Intellectual Property (1)
SP5. Legal Issues in Computing (1)
SP6. Organizational Context (2)
SP7. Professional & Ethical Issues/Responsibilities (1.5)
SP8. History of Computing (1)
SP9. Privacy and Civil Liberties (1)

WS. Web Systems and Technologies (19 hours)
WS1. Web Technologies (9)
WS2. Information Architecture (4)
WS3. Digital Media (1)
WS4. Web Development (3)
WS5. Vulnerabilities (2)
WS6. Social Software (0)

Total Number of Hours in 2-Year Model = 212
Total Number of Hours in 4-Year Model = 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.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines
Chapter 6
Overview of the Curricular Model

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 two-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 advanced positions or pursue four-year 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 two-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 two-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 fourteen program outcomes of IT programs (section 1.4, 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, and
- 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 are *integration first* and *pillars-first*. These two approaches are outlined in two subsections of the four-year Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. The two-year curriculum guide only outlines the *pillars-first* approach. The two year approach includes a description of the *pillars-first* approach including a list of course names, arranged roughly in the order that a student would take them during the two years. The full two year coursework will vary depending on the curriculum degree goal – terminal or transfer. The *pillars-first* approach provides suggested coursework that is designed for the transfer to a four year degree program.

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, Integrative Programming and System Integration could be taught in the second year of the program.

6.3.1 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 and System Administration
- Technical and Professional Communication

- Fundamentals of Web Systems and Human Computer Interaction
- Fundamentals of Information Management
- Integrative Programming and System Integration
- 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.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

Chapter 7
The Core in the 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 students to solve all types of computing and informational problems, regardless of the 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

There are some topics which should be covered in the introductory curriculum; these topics include IT fundamentals, programming fundamentals, and the IT pervasive themes.

7.2.1 IT Fundamentals

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.

Given the nature of the work that IT professionals typically do, it is strongly recommended that students be exposed to problem solving early in their education. It is recommended that a problem solving approach be adopted throughout their program in order for the students to practice the skills that will serve them daily in their chosen profession.

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 developed 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 includes 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 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.

These may include greater depth of study in any of the core areas or other subject areas as describe in the Body of Knowledge. Priority must be given to the requirements of the degree program into which the student intends to transfer.

7.4 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.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

Chapter 8
Completing the Curriculum

The primary purpose of Chapter 7 was to outline the material in the model curriculum that occurs within the first two years of study. As emphasized in the Information Technology Volume IT2008, the Information Technology core provided in IT2008 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. To the 2-year IT Volume, this can be accomplished in two ways – by students continuing their education to obtain a baccalaureate degree or by completing additional coursework to prepare them to obtain an entry level position as an Information Technology Technician. 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. The requirements in Section 8.1 address the requirements set forth in the IT Volume for baccalaureate students. Section 8.2 outlines a set of additional courses to provide employability skills for students completing a terminal associate's degree.

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 section outlines several general recommendations for associate IT programs seeking to meet these goals. Specific details of the program to which a student intends to transfer should be consulted when selecting coursework to satisfy the skills discussed in this section.

Students who decide to continue their educational experience by obtaining a baccalaureate degree that follows the IT Volume will have additional opportunities beyond their associate degrees to enhance and improve the skills discussed in this section. To this end, associate degree programs in IT should incorporate the skills in this section as time permits.

8.1.1 Computational Thinking – Problem Solving for Computing

Jeanette Wing's research on Computational Thinking identifies the following characteristics:

- *Conceptualizing, not programming*
- *Fundamental, not rote skill*
- *A way that humans, not computers think*
- *Complements and combines mathematical and engineering thinking*
- *Ideas, not artifacts*
- *For everyone, everywhere* (Wing, 2006)

Computational thinking has also been identified through algorithmic techniques that can be used in designing solutions such as thinking recursively, thinking abstractly, thinking ahead, thinking procedurally, thinking logically, and thinking concurrently (Wing, 2006).

In today's information and technology-rich 21st Century environment, students are tasked with solving complex problems in a timely and efficient manner. Employers underscore the need for workers who can think in terms of both systems and solutions – students who can work in teams to find facts, challenge assumptions and troubleshoot issues as they arise in the world of work. Problem-solving skills are in high demand for the worker of the 21st century.

Networking, web, databases, programming, human computer interaction – all part of the IT body of knowledge – must now respond to the demands of increasingly complex social systems that demand workers meet these challenges as complex problems. Students need models of how to solve messy, authentic, complex problems that they will encounter on any job. They need practice and guidance in how to work through a sequence and process of problem solving. And they need to work collaboratively to work through these problems and then build the best solutions possible using 21st Century tools and methods to get there.

8.1.2 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
- Appreciate multiculturalism, globalization, and cultural issues that arise
- Write appropriate electronic communications (including email, blogs, instant messages, virtual team correspondence, 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.

8.1.3 Working in teams

Few computing professionals can expect to work in isolation for very much 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 education.

To ensure that students have the opportunity to acquire these skills as undergraduates, we recommend that all associate degree IT programs include opportunities to work in teams, beginning relatively early in the curriculum. Students will have additional opportunities to work in teams as they continue their educational experiences toward a bachelor's degree.

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, being aware that their professional knowledge provides them with unique opportunities to contribute to a broader society and understanding the implications of both local and global 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 Topics for further study

In this context, topics for further study can be used as a means to develop coursework whose content addresses additional skills necessary to complete an applied or terminal associate degree program. Institutions will wish to orient their additional courses to their own areas of expertise, guided by the needs of students, the expertise of the faculty members, and the needs of the institution's advisory boards.

Potential topics for further study appear in Figure 8-1.

Figure 8-1.Topics for further study

Human-Computer Interaction
- Human-Centered Design and Evaluation
- Graphical User Interface
- Multimedia Systems Development
- Interactive Systems Development
- Computer-Supported Cooperative Work
- Human Cognitive Skills

Information Assurance and Security
- Forensics and Incident Response
- Biometrics
- Security Policies and Procedures
- Security Vulnerabilities

Information Management
- Information Architecture
- Database Design
- Transaction Processing
- Distributed and Object Database
- Data Mining
- Data Warehousing
- Multimedia Information Systems
- Digital Libraries

Integrative Programming & Technologies
- n-Tier Architectures
- Scripting Concepts

Networking
- Distributed Systems
- Wireless and Mobile Computing
- Cluster Computing
- Data Compression
- Network Security
- Enterprise Networking
- Digital Communications

Programming
- Object-Oriented Programming
- Event-Driven Programming
- Logic Programming
- Programming Languages

Platform Technologies
- Computer Architecture
- Parallel Architectures
- Hardware Implementation Technologies
- Computing Techniques
- Cloud Computing

Support Services
- Technical Support
- Customer Service and Support
- Support Center Design and Management

System Administration and Maintenance
- Network Management
- Database Administration

System Integration and Architecture
- Software Acquisition and Implementation
- System Needs Assessment
- Data Center Design
- Software Economics
- Enterprise Systems
- Knowledge Management
- Computing Economics
- Software Testing

Social and Professional Issues
- Professional Practice
- Social Context of Computing
- Computers and Ethics
- IT and Economic Development
- Computer Law
- Intellectual Property
- Privacy and Civil Liberties
- Globalization
- Change Catalyzed by IT
- Environmental Issues (Green Technologies)

Web Systems and Technologies
- Programming for the WWW
- E-commerce
- Data-Driven Websites
- Web Software Tools

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

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:

- *Capstone courses.* These courses typically form a one- or 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 typically 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 one or more of 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

For institutions preparing students through an IT transfer associate degree program, it is necessary to refer to the statement in the IT Volume concerning vendor and industry certifications. It should be expected that, in general, certifications will not be accepted for transfer credit to a 4-year program nor will coursework designed exclusively to prepare students to take a certification exam. That said, institutions preparing students to enter the workforce at the end of a terminal associate degree program may need to consider the expectations employers hold of their graduates with respect to specific industry certifications.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

Chapter 10
Characteristics of IT Graduates

As stated in the 4-year IT Volume, graduates of 4-year IT programs should possess the following characteristics:

(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.

Within the 2-year transfer degree, it is desirable that the foundations of these skills be introduced so that students can continue building upon them throughout their aggregate four years of study. It is recognized that different programs will have a greater or lesser ability to incorporate specific skills within these areas and that not all areas would receive uniform treatment. Most importantly the program developers should be conscientious of the needs of specific programs into which their students will transfer. It should be noted that the above mentioned skill set is predominantly defined by the ABET accreditation criterion for 4-year programs. It is expected that at such time that such criteria are developed for 2-year transfer programs, those criteria will more clearly define the expected level of achievement necessary in a 2-year degree.

The IT Volume also addresses pervasive themes that run throughout the program outcomes. As such, program developers for associate degree programs should be aware of the need to incorporate those pervasive themes within the 2-year IT curriculum. The pervasive themes follow.

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.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

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 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 specifications

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.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

Chapter 12
Institutional Challenges

Implementing a curriculum successfully 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 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 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 lifelong 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 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 racial, 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 faculty should stay current by engaging in professional development and constantly be looking for better ways to deliver the curriculum.* Constant improvement in all areas should be a hallmark of a healthy IT program.

12.2 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.

12.3 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 a 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. Departments must start the recruiting process very early and should consider reaching out to a wide range of potential applicants, including people currently working in industry.

• *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.

12.4 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.

Special Interest Group for IT Education (SIGITE)

Two-Year Information Technology Guidelines

Appendix A
The IT Body of Knowledge

This appendix to the Computing Curricula – Two-Year Information Technology Volume defines the knowledge domain that is likely to be taught in a two-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 two-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 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 could be a blueprint for a four-year transfer degree.* Because the core is defined as minimal, it does not, by itself, constitute a complete two-year curriculum.
- *Core learning outcomes are not necessarily achieved in a set of introductory courses early in the two-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 basic system, including hardware and software installation and testing, at some point during their two-year program. The material that is essential to successful management of projects at this level is therefore part of the core, since it is required of all students. At the same time, the project course experience should come at the end of a student's two-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.

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.

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 an example in which the IT body of knowledge could be packaged into courses.

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)

The Information Technology Two-Year Body of Knowledge

ITF. Information Technology Fundamentals (25 hours)
ITF1. Pervasive Themes in IT (17)
ITF2. History of Information Technology (3)
ITF3. IT and Its Related and Informing Disciplines (3)
ITF4. Application Domains (2)

HCI. Human Computer Interaction (11 hours)
HCI1. Human Factors (2)
HCI2. HCI Aspects of Application Domains (2)
HCI3. Human-Centered Evaluation (2)
HCI4. Accessibility (2)
HCI5. Emerging Technologies (2)
HCI6. Human-Centered Software Development (1)

IAS. Information Assurance and Security (13 hours)
IAS1. Fundamental Aspects (2)
IAS2. Security Mechanisms (Countermeasures) (3)
IAS3. Operational Issues (2)
IAS4. Policy (2)
IAS5. Attacks (2)
IAS6. Forensics (0.5)
IAS7. Threat Analysis Model (0.5)
IAS8. Vulnerabilities (1)

IM. Information Management (25 hours)
IM1. IM Concepts and Fundamentals (8)
IM2. Database Query Languages (8)
IM3. Data Organization Architecture (7)
IM4. Data Modeling (2)

IPT. Integrative Programming & Technologies (10 hours)
IPT1. Intersystems Communications (1)
IPT2. Data Mapping and Exchange (2)
IPT3. Scripting Techniques (4)
IPT4. Software Security Practices (1)
IPT5. Overview of Programming Languages (1)
IPT6. Miscellaneous Issues (1)

MS. Math and Statistics for IT (17 hours)
MS1. Functions, Relations and Sets (6)
MS2. Basic Logic (10)
MS3. Application of Math & Statistics to IT (1)

NET. Networking (14 hours)
NET1. Foundations of Networking (3)
NET2. Routing and Switching (7)
NET3. Physical Layer (2)
NET4. Security (1)
NET5. Network Management (0)
NET6. Application Areas (1)

PF. Programming Fundamentals (35 hours)
PF1. Fundamental Data Structures (7)
PF2. Fundamental Programming Constructs (10)
PF3. Object-Oriented Programming (9)
PF4. Algorithms and Problem-Solving (6)
PF5. Event-Driven Programming (3)

PT. Platform Technologies (12 hours)
PT1. Operating Systems (8)
PT2. Architecture and Organization (3)
PT3. Computing Infrastructures (1)

SA. System Administration and Maintenance (6 hours)
SA1. Operating Systems (3)
SA2. Applications (1)
SA3. Administrative Activities (2)

SIA. System Integration and Architecture (8.5 hours)
SIA1. Requirements (4)
SIA2. Integration and Deployment (1)
SIA3. Project Management (1)
SIA4. Testing and Quality Assurance (2)
SIA5. Organizational Context (0.5)

SP. Social and Professional Issues (16.5 hours)
SP1. Professional Communications (2)
SP2. Teamwork Concepts and Issues (5)
SP3. Social Context of Computing (2)
SP4. Intellectual Property (1)
SP5. Legal Issues in Computing (1)
SP6. Organizational Context (2)
SP7. Professional & Ethical Issues/Responsibilities (1.5)
SP8. History of Computing (1)
SP9. Privacy and Civil Liberties (1)

WS. Web Systems and Technologies (19 hours)
WS1. Web Technologies (9)
WS2. Information Architecture (4)
WS3. Digital Media (1)
WS4. Web Development (3)
WS5. Vulnerabilities (2)
WS6. Social Software (0)

Total Number of Hours in 2-Year Model = 212
Total Number of Hours in 4-Year Model = 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.

Model Curriculum Core Learning Outcomes

IT Fundamentals (ITF) – 25 core hours

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 1. Pervasive Themes in IT

Minimum core coverage time: 17 hours

Topics:

- User centeredness and advocacy
- IT professional roles as problem solver and solutions provider
- Information assurance and security
- IT systems model
- Management of Complexity (abstraction, modeling, best practices, patterns, standards, prioritization, and use of appropriate 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. Describe the components of IT systems and their interrelationships.
2. Describe how complexity occurs in IT.
3. Recognize that an IT professional must know how to manage complexity.
4. List examples of tools and methods used in IT for managing complexity.
5. Describe the importance of prioritizing competing needs for IT solutions.
6. Describe the roles of the IT professional as a user advocate, problem solver and solutions provider.
7. Explain why life-long learning and continued professional development is critical for an IT professional.
8. Explain why adaptability and interpersonal skills are important to an IT professional.
9. Distinguish between data and information, and describe the interrelationship.
10. Describe the importance of data and information in IT.
11. Explain the importance of using information and communication technologies to solve problems.

12. Explain why the information assurance and security perspective needs to pervade all aspects of IT.
13. Identify how organizational context influences and impacts the development and deployment of IT systems.

ITF2. 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 each other.

ITF3. 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

Core learning outcomes:

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

ITF4. 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
(and others)

Core learning outcomes:

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

Human-Computer Interaction (HCI) – 11 core hours

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.

HCI1. Human Factors
Minimum core coverage time: 2 hours

Topics:

Understanding the user
Designing for humans – e.g., conceptual models, feedback, constraints, mapping, stages of action
Ergonomics

Core learning outcomes

1. Distinguish among different user populations with regard to their abilities and characteristics for using both software and hardware products.
2. Explain the importance of user abilities and characteristics in the usability of products.

Advanced learning outcomes:

1. Design a product for a specific user population.
2. Explain how the physical aspects of product design impact its usability.

HCI2. HCI Aspects of Application Domains
Minimum core coverage time: 2 hours

Topics:

Types of environments
Approaches

Core learning outcomes

1. Describe different types of interactive environments.
2. Describe the differences between developing a user interface for a web page and a standalone application.
3. Describe several characteristics of a web environment that can enhance the usability of a web-based application.
4. Discuss the relationship of the user's application domain to the development of an appropriate user interface.

Advanced learning outcomes:
1. Develop user interfaces for domain specific applications.
2. Interview domain experts to capture domain specific requirements for user interfaces.

HCI3. Human-Centered Evaluation
Minimum core coverage time: 2 hours

Topics:
Usability

Core learning outcomes
1. Perform a simple usability evaluation for an existing software application.
2. Describe common usability guidelines.

HCI4. 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. Discuss the use of accessibility features, such as a narrator.

HCI5. Emerging Technologies
Minimum core coverage time: 2 hours

Topics:
- Alternative input/output devices
- Alternative displays (heads-up, goggles, etc.)
- Mobile computing
- Wearable computing
- Pervasive computing

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. Describe the advantages of emerging I/O devices as compared to current devices.
2. Describe the ideal characteristics of a wearable display and compare these to what can be done with today's technology.

HCI6. Human-Centered Computing
Minimum core coverage time: 1 hour

Topics:
User-centered design methods
Software development lifecycles

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.

Information Assurance and Security (IAS) – 13 core hours

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.

IAS1. Fundamental Aspects
Minimum core coverage time: 2 hours

Topics:
History and Terminology
Security Mindset (reasoned paranoia)
Design Principles (Defense in Depth)
System/security life-cycle
Disaster recovery (natural and man-made)
Digital 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. Describe the security mindset and the role of "paranoia" in that mindset.
4. Explain and give examples of why security and assurance must be "built in" to design and architecture from the beginning to be most effective.
5. Outline the system life-cycle and its relationship to security.
6. Distinguish among information states such as transmitting data, storing data and processing data.

7. Describe a simple disaster recovery scenario.
8. Define digital forensics.
9. Describe a situation where a digital forensic investigation would be necessary.

IAS2. Security Mechanisms (Countermeasures)

Minimum core coverage time: 3 hours

Topics:

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

Core learning outcomes

1. Describe 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. Describe the characteristics of an effective password.
4. Describe and compare physical access control to logical access control.
5. Identify the key types of biometric information utilized in authentication.
6. Describe the differences between symmetric and asymmetric cryptosystems.
7. Define integrity, confidentiality, authentication and non-repudiation.
8. Describe digital signatures and certificates.
9. Define public key infrastructure (PKI).

Advanced learning outcomes:

1. Describe the single sign-on authentication process and problems related to using and implementing this technology.
2. Compare key access control and authentication mechanisms (Kerberos, RAS, etc.).
3. Compare the advantages and disadvantages of centralized access controls to decentralized access controls.

IAS3. Operational Issues

Minimum core coverage time: 2 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. Describe the importance of and key elements involved in incident tracking.
3. Identify risks associated with disasters or disruptions and specify key mitigation strategies.
4. Identify the types of company assets to be protected by a security plan.
5. Specify the key aspects of physical site security.
6. Describe acceptable use policies and practices that are relevant to safeguarding an organization's information assets.

Advanced learning outcomes:

1. Describe how key information security and assurance standards are or should be utilized in specific industry contexts.
2. Discuss the role of CASPR (Commonly Accepted Security Practices and Recommendations) forms in defining and approving standard operational and management practices.
3. Specify how changes in technology and the constantly changing methods of attacking systems and attempting to compromise information or access it inappropriately impact operational and managerial practices and policies.

IAS4. Policy
Minimum core coverage time: 2 hours

Topics:

Importance of Policies
Prevention
Avoidance
Incident Response (Forensics)

Core learning outcomes:

1. Explain why policies and procedures are necessary for security.
2. Describe why a strong password policy is important.
3. Describe why a password policy might need to be modified due to changing circumstances.
4. Explain why security policies must consider all aspects of an organization in order to be effective.
5. Describe a situation in which an incident would require a full forensic approach and how failure to follow good forensic procedures could make prosecution impossible.
6. 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.

IAS5. 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. List social engineering techniques used to gain access to computing and network assets in an organization.
2. Describe how a Denial of Service attack works against an organization's network.
3. List some different protocol attacks to which TCP/IP is susceptible.
5. Discuss techniques used during an active attack.
6. Discuss techniques used during a passive attack.
7. Describe how an active attack might use information from a passive attack to compromise a system.
8. Describe 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).

IAS6. Forensics
Minimum core coverage time: 0.5 hour

Topics:
Rules of Evidence
Search and Seizure
Digital Evidence

Core learning outcomes:

1. Identify the proper steps in a full forensic incident response.

IAS7. Threat Analysis Model
Minimum core coverage time: 0.5 hour

Topics:
Risk assessment

Core learning outcomes:

1. Identify the aspects of a business that may be impacted by a security breach or interruption of operation.

IAS8. Vulnerabilities
Minimum core coverage time: 1 hour

Topics:
Perpetrators
Internal attacks
External attacks
Ignorance
Carelessness
Network
Hardware & software vulnerabilities

Core learning outcomes:

1. Explain the differences between a hacker and a cracker.
2. Describe the role of the user in information assurance and how they fit into an overall information assurance plan for an organization.
3. Explain how ignorance and carelessness leads to vulnerabilities for an organization.
4. Identify similarities and differences between internal and external attacks.
5. List the major network threats and vulnerabilities common to organizations.
6. Discuss how software can contribute to the vulnerabilities for an organization.
7. Discuss how hardware can contribute to the vulnerabilities for an organization.

Advanced learning outcomes:

1. Describe how you would test a system for vulnerabilities.
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.

Information Management (IM) – 25 core hours

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.

IM1: 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 key terms such as information, data, database, database management system, metadata.
2. Explain the role of data, information, and databases in organizations.
3. Explain how data storage and retrieval has evolved with technology.
4. Explain the advantages of the database approach over traditional file processing.
5. Identify and explain the general types of databases: personal, workgroup, department, and enterprise.
6. Describe 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, e.g., automated data collection, input forms, source documents, and discuss their implications.
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.
11. Design and create a simple database.

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.

IM2: Database Query Languages:
Minimum core coverage time: 8 Hours

Topics:

SQL data manipulation
SQL data definition
Reports
Query by example

Core learning outcomes:
1. Create and test SQL queries using SELECT, FROM, WHERE, and ORDER BY blocks.
2. Use logical and set operators, such as UNION, DISTINCT, LIKE, and BETWEEN.
3. Formulate and test queries using aggregate functions with GROUP BY and HAVING clauses.
4. Formulate and test queries using sub-queries, VIEWS and joins.
5. Create and format output reports (e.g., header, footer, totals, subtotals) using SQL options and post-processing features of an SQL environment.
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. Formulate and test queries using query by example.
8. Use report generators.

Advanced learning outcomes:
1. Use nested correlated sub-queries with (NOT) EXISTS options.
2. Use substitution variables, SEQUENCES, and OUTER joins.
3. Define referential integrity constraints and DELETE/INSERT/UPDATE options SET NULL, SET DEFAULT, CASCADE, and RESTRICT (NO ACTION).

IM3: 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
- 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

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.

IM4. Data Modeling
Minimum core coverage time: 2 hours

Topics:
- Conceptual Models
  - Entity Relationship diagrams
  - Identification of business rules
- Logical Models
- Physical Models

Core learning outcomes:
1. Describe and interpret Entity Relationship diagrams.
2. Create a simple Entity Relationship diagram.
3. Identify business rules.
4. Describe a logical model.
5. Describe a physical model.

Advanced learning outcomes:
1. Formulate and explain business rules.
2. Create and evaluate a logical model.
3. Create and evaluate a physical model.

Integrative Programming and Technologies (IPT) – 10 core hours

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.

IPT1. Intersystem Communications
Minimum core coverage time: 1 hour

Topics:
- Architectures for integrating systems

Core learning outcomes:
1. Describe and contrast the different types of architectures for integrating systems.

IPT2. Data Mapping and Exchange
Minimum core coverage time: 2 hours

Topics:
- Metadata
- Data representation and encoding
- XML and Document Object Model

Core learning outcomes:
1. Define metadata.

2. Describe the characteristics of data encoding schemes such as ASCII, EBCDIC, and Unicode.
3. Describe tools such as XML and the Document Object Model used to integrate and exchange data between systems.

IPT3. Scripting Techniques
Minimum core coverage time: 4 hours

Topics:

Scripting and the role of scripting languages
Creating and executing scripts

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.

IPT4. Software Security Practices
Minimum core coverage time: 1 hour

Topics:

Best security coding practices

Core learning outcomes:

1. Define the goals of secure coding.
2. For each of the following, give an example of a problem that can occur when best practices are not followed and then describe how to overcome the problem:
    a. Validation of user input.
    b. Error handling

IPT5. Overview of Programming Languages
Minimum core coverage time: 1 hour

Topics:

Programming paradigms
Compiled vs. interpreted languages
Application vs. scripting languages

Core learning outcomes:

1. Contrast the structured and object-oriented programming paradigms.
2. Diagram and label models of compiled and interpreted programs.
3. Give an example where an application language and a scripting language would be most appropriate, and justify each choice.

IPT6. Miscellaneous Issues
Minimum core coverage time: 1 hour

Topics:

Versioning and version control

Core learning outcomes:

1. Tell why it is important to version software and describe one mechanism that can be use to control the versioning of software.

Mathematics and Statistics for IT (17 core hours)

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 obtain. 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 acquire. 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 elective learning outcomes in detail.

MS1. 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.

MS2. 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.

MS3. Application of mathematics to IT

Minimum core coverage time: 1 hour

Topics:

Math in IT

Core learning outcomes

1. Explain, with examples, the importance of a range of mathematical concepts, including sets, relations, functions, and basic logic for IT.

Networking (NET) – 14 core hours

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.

NET1. 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 impacts 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.

NET2. Routing and Switching
Minimum core coverage time: 7 hours

Topics:

IEEE 802.1
Routing protocols
Device architecture
Routing and switching
Latency, response time, jitter

Core learning outcomes:

1. Summarize and describe a data communications model, protocol, standard, and architecture in use today.
2. Discuss the concepts and the "building blocks" of today's data communication networks such as switches, routers, and cabling.
3. Describe the operation of various network devices as defined in IEEE standards for network components.
4. Describe the necessary hardware (switches and routers) and components (routing algorithms and protocols) used to establish communication between multiple networks.
5. Discuss 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. Connect two networks.
2. Explain the operation and function of 802.1 devices and protocols.
3. Analyze and explain routing algorithms and protocols, process routing tables and configure routers for proper operation.
4. Describe the effect of various topologies, applications, and devices on network performance topics such as latency, jitter, response time, connection loss, and quality of service.
5. Compare and contrast routing protocols and compare/contrast the functions and operation of interior routing protocols with exterior routing protocols.
6. Explain vlans.
7. Illustrate how load balancing is accomplished in routers and switches and deploy and test devices utilizing load balancing.
8. Describe strategies to ensure the availability of network access in switched and routed networks.
9. Select appropriate routing and switching equipment for a given network application.

NET3. Physical Layer
Minimum core coverage time: 2 hours

Topics:

Wireless & mobile links
Physical media
Error detection and correction
Communication standards
IEEE 802
Topologies

Core learning outcomes:

1. List several types of physical communication media, and compare their bandwidth characteristics.
2. Describe the physical challenges inherent to wireless-fixed and wireless-mobile communication channels.
3. Identify methods of error handling such as parity, CRC, and EDC.
4. Describe how most modern communication standards are developed, addressing both *de jure* and *de facto* standards.
5. 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. Select appropriate physical media for a given network application.

NET4. Security
Minimum core coverage time: 1 hour

Topics:

VPN applications
Firewalls
Wired, wireless & mobile

Core learning outcomes:

1. Define public and private keys.
2. Define SSL.
3. Describe a remote access VPN.
4. Describe a point-to-point VPN.
5. Describe a firewall.
6. Describe a scenario where an IDS could detect a password cracking attempt.
7. List security concerns related to wired, wireless, and mobile networking.

Advanced learning outcomes:

1. Install and configure a firewall.
2. Demonstrate that a firewall is properly configured using a vulnerability testing tool.

NET5. Network Management
Minimum core coverage time: 0 hours

Topics:
Wireless & mobile
Wired
Security

Core learning outcomes:
None. The coverage time for everyone is 0 hours.

Advanced learning outcomes:
1. Design and implement either a local area or wide area network.
2. Configure the network nodes (computers, routers, etc).
3. Troubleshoot a network problem.
4. Develop a backup and disaster recovery plan as it relates to the network components of an organization.
5. Use a network management tool to collect performance data from a set of network nodes.

NET6. Application Areas
Minimum core coverage time: 1 hour

Topics:
World Wide Web
Database and file services

Core learning outcomes:
1. Describe what would happen to the World-wide web portion of the Internet if the majority of all routers ceased to function.
2. Describe the role of networking in database and file service applications.

Programming Fundamentals (PF) – 35 core hours

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.

PF1. Fundamental Data Structures
Minimum core coverage time: 7 hours

Topics:
Primitive types
Arrays
Records
Strings and string processing
Pointers and references
Strategies for choosing the right data structure

Core learning outcomes:
1. Discuss the use of 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.
4. Choose the appropriate data structure for modeling a given problem.

PF2. 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. 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.
7. Describe the concept of recursion and give examples of its use.

PF3. Object-Oriented Programming
Minimum core coverage time: 9 hours

Topics:

Object-oriented design
Encapsulation and information hiding
Separation of behavior and implementation
Classes and objects
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. Explain the relationship between the static structure of the class and the dynamic structure of the instances of the class.
7. Describe how iterators access the elements of a container.
8. Describe how constructors and destructors relate to the life of an object.
9. Identify and discuss the concept of a class and an object.

PF4. 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 an algorithm for solving a simple problem.
4. Use pseudo-code or a programming language to implement, test, and debug an algorithm for solving a simple problem.
5. Describe strategies that are useful in debugging.

PF5. 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 a simple event-driven program that responds to user events.
3. Develop code that responds to exception conditions raised during execution.

Platform Technologies (PT) – 12 core hours

IT professionals will encounter a variety of platforms in their career. 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.

PT1. Operating Systems
Minimum core coverage time: 8 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

Core learning outcomes:

1. Define the necessary components and functions of an operating system.
2. Compare at least two operating systems and describe their suitability to a given task or goal.
3. Install a current operating system and validate that the installation was successful.
4. Describe the similarities and differences between Windows and Unix-class systems.
5. Explain the benefits of using scripts to perform operating systems tasks.

Advanced learning outcomes:

1. Analyze operating system requirements and recommend an appropriate operating system to meet the requirements.
2. Install a second operating system and validate that the installation was successful.
3. Write at least one script to perform an operating system task.

PT2. 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
- Firmware, software, and hardware

Core learning outcomes:

1. Describe how numbers and characters are represented in a computer.
2. Interpret 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.
5. Distinguish between firmware, software, and hardware.

Advanced learning outcomes:

1. Interpret low-level code that takes two numbers from the keyboard, adds them in binary, and sends the result to the screen.

PT3. 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. Describe the need for power and heat budgets within an IT environment.
3. List the various types of servers 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. Deliver computing services on a network.
3. List the benefits of using server farms.

System Administration and Maintenance (SA) – 6 core hours

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.

SA1. Operating Systems
Minimum core coverage time: 3 hours

Topics:

Installation
Configuration
Maintenance (service packs, patches)
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. Describe the importance of system maintenance for an organization.
3. Identify situations in which a system needs to be reconfigured.
4. Recognize when a system requires maintenance.
5. Distinguish between server and client services.
6. Describe open source and proprietary software.

Advanced learning outcomes:
1. Compare various operating systems and recommend a particular operating system to satisfy given needs.
2. Modify the configuration of an operating system.
3. Install service packs and operating system patches.
4. Install a server and client service.

SA2. Applications
Minimum core coverage time: 1 hour

Topics:
Installation
Configuration

Core learning outcomes:
1. Install at least one current application.
2. Discuss the benefits of custom configuration of applications.
3. Describe open source and proprietary software.

Advanced learning outcomes:
1. Install patches for applications.

SA3. Administrative Activities
Minimum core coverage time: 2 hours

Topics:
Resource management
Backup management
Security management
Disaster recovery
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 of IT systems.
5. Explain why users need to be trained on IT systems and policies.

System Integration and Architecture (SIA) – 8.5 core hours

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, source, evaluate and integrate components into a single system, and validate the system. It also covers the fundamentals of project management and the interplay between IT applications and organizational processes.

SIA1. Requirements
Minimum core coverage time: 4 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. Describe 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 how requirements gathering fits into a system development lifecycle.

SIA2. Integration and Deployment
Minimum core coverage time: 1 hour
Topics:

- Components, interfaces, and integration
- Infrastructure, Middleware, and Platforms

Core learning outcomes:
1. Define integration in terms of components and interfaces.
2. Give an example of a middleware platform.
3. List advantages and disadvantages of a middleware platform.

SIA3. Project Management
Minimum core coverage time: 1 hour

Topics:

- Project plan components
  - Roles/Responsibilities/accountability
  - Finance/estimation/budgeting
  - Planning
  - Risk analysis
  - Scheduling
  - Tracking
  - Post-mortems

Core learning outcomes:
1. Describe the key components of a project plan.

Advanced learning outcomes:
1. Working on a team, participate in the creation of a project plan.

SIA4. Testing and Quality Assurance

Minimum core coverage time: 2 hours

Topics:

Techniques
Usability
Acceptance criteria

Core learning outcomes:

1. List the various components of usability testing.
2. Describe one technique used in testing a system or product.
3. List possible acceptance criteria.

Advanced learning outcomes:

1. Summarize the data from a usability test and make appropriate recommendations.
2. Write a set of system tests based on use cases.

SIA5: Organizational Context

Minimum core coverage time: 0.5 hour

Topics:

Business processes

Core learning outcomes:

1. Describe the relationship between business processes and system integration.

Social and Professional Issues (SP) – 16.5 core hours

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.

SP1. Professional Communications

Minimum core coverage time: 2 hours

Topics:

Oral presentations
Technical documents

Core learning outcomes:

1. Prepare and present an oral presentation for a user audience.
2. Create an oral presentation for a management audience.
3. Create a basic technical document.

SP2. 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.

SP3. Social Context of Computing
Minimum core coverage time: 2 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. Describe positive and negative ways in which information technology alters the modes of interaction between people.
2. Explain why computing and networking access is restricted in some countries.
3. Explain what the digital divide is and why it has developed.
4. Identify underlying gender, cultural and diversity issues in information technology.
5. Identify how information technology changes and affects culture as a whole.
6. Identify how the Internet has changed the face of computing and how it has affected society.

SP4. Intellectual Property
Minimum core coverage time: 1 hour

Topics:
Foundations of intellectual property
Ownership of information
Plagiarism
Software piracy
Fair use
Digital Millennium Copyright Act (DMCA)

Copyrights, patents, trademarks and trade secrets

Core learning outcomes:
1. Distinguish among copyrights, patents, trademarks and trade secrets.
2. Discuss the implications of plagiarism both in education and the profession.
3. Discuss the consequences of software piracy on information technology and the role of relevant enforcement organizations.
4. Describe consequences of the Digital Millennium Copyright Act.

SP5. Legal Issues in Computing
Minimum core coverage time: 1 hour

Topics:
Compliance (ADA508, FERPA, HIPPA, Sarbanes-Oxley…)
Hackers/crackers
Computer crime
Viruses
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, and HIPPA.
5. Describe the differences in accountability, responsibility, and liability.

SP6. 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. Identify when an organizational culture can affect IT.

SP7. Professional and Ethical Issues & Responsibilities
Minimum core coverage time: 1.5 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 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 codes in assignments.
4. Identify a whistle-blowing incident.
5. List one underlying philosophy 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. Discuss the pros and cons of ethical hacking.

SP8. 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 an emerging technology in the context of the history of computing technologies.
2. Identify significant continuing trends in the information technology profession.
3. Identify how life-long learning impacts the information technology professional.

SP9. Privacy and Civil Liberties
Minimum core coverage time: 1 hour

Topics:

Privacy and civil liberties laws
HIPPA and FERPA
E.U. Data Protection
Gramm-Leach-Bailey Act

Core learning outcomes:

1. Identify issues of privacy and civil liberties in computing.
2. Discuss legislation in regards to privacy and civil liberties in computing.

Web Systems & Technologies (WS) – 19 core hours

IT applications are increasingly web-based, incorporate a variety of media types, and involve multiple users. Diverse multi-cultural and multi-lingual user communities have emerged because of the Web. This knowledge area covers the design, implementation, and testing of web-based applications and social software, and the incorporation of a variety of digital media into these applications. It also covers social, ethical, and security issues arising from the Web and social software.

WS1. Web Technologies

Minimum core coverage time: 9 hours

Topics:

HTTP Protocol
Presentation technologies
Web-markup and display languages
Emerging technologies
Standards & Standard 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. Use XML syntax to create a document in a Web application.
5. Use a presentation technology, such as Cascading Style Sheets and DHTML.
6. Describe data entry and validation techniques in client-side vs. server-side programming.
7. Identify client-side and server-side security issues.
8. Describe the use of server-side backend databases in web sites and web applications.
9. Describe current and emerging technologies used in Web Services including open source languages and packages, proprietary languages and packages, and enterprise Web development and distributed Web applications.
10. Discuss Web standards and standard bodies including the World Wide Web Consortium (W3C).

Advanced learning outcomes

1. Apply HTML/XHTML/XML syntax to create documents.
2. Apply HTML/XHTML/XML syntax to generate contents via programming.
3. Apply XML syntax to transform documents between formats.
4. Construct website to include client-side programming such as JavaScript, Java Applets, Flash, and other Web GUI technologies.
5. Discuss data persistence via cookies in maintaining states.
6. Write a web application that uses server-side programming.
7. Identify issues related to server-side security.
8. Discuss connectivity issues with backend databases.
9. Implement Web solutions that comply with Web standards and standard bodies including specifications, guidelines, software, and tools.

WS2. Information Architecture
Minimum core coverage time: 4 hours

Topics:

- Hypertext/hypermedia:
  - Effective communication
  - Interfaces
  - Navigation schemes
  - Media types
- Web design process:
  - User-driven design
  - Web Design Patterns
  - Information organization
  - Usability

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. Select an organization of information based on its inherent structure (e.g., chronological, alphabetic).
4. Identify the purpose of a web site or genre (e.g., ecommerce, self-service, educational, governmental service).
5. Choose a graphic file type that matches the image characteristics and use.
6. Identify time-based media types commonly used on the web (e.g., Flash, streaming media).
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. List user characteristics that affect web site design.
10. List 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 navigational framework that match the content and genre of the site.
5. Create page templates to maintain style consistency and branding, and to simplify development of a site.
6. Design an effective homepage for a site.
7. Create client-side web-based user interfaces for navigation.
8. Create server-side web-based user interfaces for applications.

WS3. Digital Media
Minimum core coverage time: 1 hour

Topics:

- Digital Libraries
- Media formats
- Capture, Authoring and Production Tools

Core learning outcomes:
1. Identify and access a major digital library.
2. Use a media acquisition tool to capture, digitize, and sample media contents.
3. Describe media acquisition tools for and techniques for multimedia authoring.
4. Identify the issues involved in deploying/serving media content.

Advanced learning outcomes:
1. Apply media acquisition tools and techniques in capturing and digitizing media contents.
2. Use a multimedia authoring tool.

WS4. Web Development
Minimum core coverage time: 3 hours

Topics:

Accessibility Issues
Web Accessibility Initiative

Core learning outcomes:
1. Explain why accessibility issues are an important consideration in web page development.
2. List some of the organizations that have developed standards for web accessibility.
3. Design and implement a web page 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.

WS5. Vulnerabilities
Minimum core coverage time: 2 hours

Topics:

Client Security
- Cookies
- Phishing
- Transaction security – certificates and secure connections
- Spyware
- Viruses
- Man-in-the-middle attacks

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. Identify common server-side configuration issues that affect security.
3. Define man-in-the-middle attacks.
4. Clean an infected system of viruses and spyware.

WS6. Social Software

Minimum coverage time: 0

Topics:

- Asynchronous and synchronous communication modalities
- Collaborative and community modalities
- Ethical issues
  - Digital Divide
  - Freedom of Speech vs. Hate Speech, Pornography
  - Privacy
  - Copy/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. Describe 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. Describe the various ethical issues associated with the web, including the digital divide, issues concerning race and gender, freedom of speech, privacy, and copy and digital content rights.

Special Interest Group for IT Education (SIGITE)
Two-Year Information Technology Guidelines

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).

In Chapter 6, it was suggested that the pillars-first approach was the best approach to facilitate transfer into a four-year program; therefore, this outline focuses on that approach.

Pillars-First Approach[1]: Introductory Courses

Title	Description	Prerequisites	KAs Covered	Units Covered
IT Fundamentals	Introduces students to the academic discipline of IT as well as the general meaning of IT.	None	ITF: all MS: (1) SIA: (1) SP: (3) 40.5 core hours	ITF1: Pervasive Themes in IT ITF2: History of IT ITF3: Related & Informing Disciplines ITF4: Application Domains MS3: Applications of Math & Stats to IT SIA5: Organizational Context SP3: Social Context of Computing SP6: Organizational Context SP8: History of computing
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 35 core hours	PF1: Fundamentals of Data Structures PF2: Fundamentals Programming Constructs PF3: Object-Oriented Programming PF4: Algorithms and Problem-Solving PF5: Event-Driven Programming
Fundamentals of Networking and System Administration	Introduces students to the fundamentals of networks and networking in IT, system administration and maintenance, and platform technologies. Includes routing, switching, physical layer, security, and application areas. Also includes operating systems, applications, administrative activities, computer architecture and organization, and computing infrastructures.	None	NET: all PT: all SA: all 32 core hours	NET1: Foundations of Networking NET2: Routing and Switching NET3: Physical Layer NET4: Security NET6: Application Areas PT1: Operating Systems PT:2 Architecture and Organization PT3: Computing Infrastructures SA1: Operating Systems SA2: Applications SA3: Administrative Activities
Technical and Professional Communication	Introduces teamwork concepts, group dynamics, leadership styles, technical writing and documentation, presentation development and delivery.	College English. Possibly offered in course offered by the English Department.	SP: (2) 7 core hours	SP1: Professional Communications SP2: Teamwork Concepts and Issues

[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
Fundamentals of Web Systems and Human Computer Interaction	Introduces students to web systems and technologies and HCI. Includes information architecture, digital media, web development and vulnerabilities of web systems. Also includes human factors, HCI aspects of application domains, human-centered evaluation, developing effective interfaces, accessibility, emerging technologies, human –centered software development	Possibly Programming Fundamentals; Possibly None. (depends on what faculty expect)	WS: all HCI: all 29 core hours	WS1: Web Technologies WS2: Information Architecture WS3: Digital Media WS4: Web Development WS5: Vulnerabilities HC1I: Human Factors HCI2: HCI Aspects of Application Domains HCI3: Human-Centered Evaluation HCI4: Accessibility HCI5: Emerging Technologies HCI6: Human-Centered Computing
Fundamentals of Information Management	Introduces students to databases and information management. Includes query languages, data organization architecture, data modeling.	Programming Fundamentals	IM: all 25 core hours	IM1: IM Concepts and Fundamentals IM2: Database Query Languages IM3: Data Organization Architecture IM4: Data Modeling
Integrative Programming and System Integration	Introduces students to integrative programming and system integration. 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 SIA: all 18.5 core hours	IPT1: Intersystems communications IPT2: Data Mapping and Exchange IPT3: Scripting Techniques IPT4: Software Security Practices IPT5: Overview of Programming Languages IPT6: Miscellaneous Issues SIA1: Requirements SIA2: Integration and Deployment SIA3: Project Management SIA4: Testing and Quality Assurance SIA5: Organizational Context
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.	Depends on the approach to this course. Possibilities include: Programming Fundamentals; Fundamentals of Web Systems; Fundamentals of Information Management	IAS: all SP: (4) 17.5 hours	IAS1: Fundamental Aspects IAS2: Security Mechanisms IAS3: Operational Issues IAS4: Policy IAS5: Attacks IAS6: Forensics IAS7: Threat Analysis Model IAS8: Vulnerabilities SP4: Intellectual Property SP5: Legal Issues in Computing SP7: Professional and Ethical Issues and Responsibilities SP9: Privacy and Civil Liberties

Bibliography

ABET (Accreditation Board of Engineering and Technology). Available at http://www.abet.org/accreditation.htm/.

ACM Code of Ethics, 1992. Available at http://www.acm.org/about/code-of-ethics.

ACM Two-Year College Education Committee, 2000. *Guidelines for Associate Degree Programs to Support Computing in a Networked Environment.* Available at http://www.acmtyc.org/reports/acmguide.pdf.

ACM Two-Year College Education Committee, 2009. *Guidelines for Associate-Degree Transfer Curriculum in Computer Science.* Available at http://www.acmtyc.org/reports/2009ComputerScienceTransferGuidelines.pdf.

American Association of Community Colleges (AACC), Available at http://www.aacc.nche.edu/AboutCC/Trends/Pages/default.aspx.

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.

CSAB (Computer Science Accreditation Board). Available at http://www.csab.org/.

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/.

DIAMOND, R. 1998. *Designing and Assessing Courses and Curricula: A Practical Guide, 5th Edition*. Jossey-Bass Publishers, San Francisco, CA.

IEEE Code of Ethics, 2006. Available at http://www.ieee.org/portal/pages/iportals/aboutus/ethics/code.html.

Joint Task Force for Computing Curricula 2001: *Computer Science*. A cooperative project of ACM and IEEE. Available at http://www.acm.org/education/education/education/curric_vols/cc2001.pdf.

Joint Task Force for Computing Curricula, 2005. Computing Curricula 2005: *The Overview Report*. A cooperative project of ACM, AIS, and IEEE-CS. Available at http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf.

Joint Task Force of the ACM and IEEE (2008). IT Volume: Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. Available at http://www.acm.org//education/curricula/IT2008%20Curriculum.pdf.

SEEPP Code of Ethics, 1999. Available at http://www.acm.org/about/se-code.

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, SanFrancisco, CA.

Wing, Jeanette (2006). *Computational Thinking,* PowerPoint presentation, http://www.imageofcomputing.com/pdf/Computational_Thinking.pdf.

Wing, Jeanette (2006). *Computational Thinking*, Communications of the ACM, March 2006, Vol.49, No. 3, p. 34.