



Computing Curricula Information Technology Volume

Version: October 2005

Table of Contents

[Executive Summary](#)

[Acknowledgements](#)

[Chapter 1. Introduction](#)

[Chapter 2. Lessons from Past Reports](#)

[Chapter 3. Changes in the Information Technology Discipline](#)

[Chapter 4. Principles](#)

[Chapter 5. Overview of the IT Body of Knowledge](#)

[Chapter 6. Overview of the Curricular Models](#)

[Chapter 7. The Core in the Curriculum](#)

[Chapter 8. Completing the Curriculum](#)

[Chapter 9. Professional Practice](#)

[Chapter 10. Characteristics of IT Graduates](#)

[Chapter 11. Computing across the Curriculum](#)

[Chapter 12. Institutional Challenges](#)

[Bibliography](#)

[Index](#)

[Appendix A. The IT Body of Knowledge](#)

[Appendix B. IT Course Descriptions](#)

[Appendix C. Learning Outcomes/Knowledge Areas Matrix](#)



Computing Curricula Information Technology Volume

Executive Summary

This document represents the final report of the SIGITE Curriculum Committee 2005 project (IT2005) – an undertaking of SIGITE (Special Interest Group on Information Technology Education) of the ACM (Association for Computing Machinery), for baccalaureate programs in Information Technology. This report dates back to December 2001, as described in [Chapter 2](#).

This volume of the report outlines a set of recommendations for baccalaureate programs in Information Technology. As described in [Chapter 1](#), the full Computing Curriculum 2004 report (CC2004) consists of several volumes, each containing separate recommendations for computing disciplines, including Computer Engineering, Computer Science, Information Technology, Information Systems, and Software Engineering. Each of these separate volumes of CC2004 is under the control of separate committees and has been or is being published as it is completed.

Highlights of this report include the following:

- *The IT body of knowledge.* We have identified a body of knowledge appropriate to baccalaureate Information Technology programs. Drawing on the structure of earlier curriculum reports (most notably the CS2001 volume), we have arranged that body of knowledge hierarchically, subdividing the field into knowledge areas, which are then broken down further into units, which are defined in terms of individual topics and learning outcomes. An overview of the body of knowledge appears in [Chapter 5](#).
- *Learning outcomes.* For each unit in the body of knowledge, we have developed a set of learning outcomes. These learning outcomes appear as part of the detailed description of the body of knowledge in [Appendix A](#). In addition to the individual learning outcomes, the report outlines a set of characteristics that all Information Technology graduates should possess in [Chapter 10](#).
- *The IT core.* From the 81 units in the body of knowledge, we have selected 76 that represent the core material, accounting for approximately 281 hours of instruction. As noted in our statement of principles in [Chapter 4](#), we defined the core as the set of units for which there is a broad consensus that the material is essential to a baccalaureate degree in Information Technology. The philosophy behind the definition of the core is described in more detail in [Chapter 5](#).
- *The IT electives.* The core is not a complete curriculum, and must be supplemented by additional material. This document proposes IT elective learning outcomes that may be used to complete a curriculum.
- *Curriculum models.* The report identifies four approaches to undergraduate instruction in Information Technology, as described in [Chapter 6](#). Building on that foundation, [Chapter 7](#)

describes the core material of the Information Technology curriculum, and [Chapter 8](#) describes the additional material necessary to constitute a complete baccalaureate curriculum in Information Technology.

- *Course descriptions.* Appendix B contains a detailed model of one curricular approach, including course descriptions for 13 courses.

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

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



Computing Curricula Information Technology Volume

Acknowledgements

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

Joseph Ekstrom	Brigham Young University	Provo, UT
Sandra Gorka	Pennsylvania College of Technology	Williamsport, PA
Reza Kamali	Purdue University – Calumet	Calumet, IN
Eydie Lawson	Rochester Institute of Technology	Rochester, NY
Barry Lunt	Brigham Young University	Provo, UT
Jacob Miller	Pennsylvania College of Technology	Williamsport, PA
Han Reichgelt	Georgia Southern University	Statesboro, GA

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

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

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

ACM Support

The IT 4-Year Curriculum Committee and the IT 4-Year Curriculum Writing Subcommittee would like to thank the ACM and their representative Russell Shackelford for generous support. Their providing of a substantial grant has made much of this work possible.



Computing Curricula Information Technology Volume

Chapter 1 Introduction

In the Fall of 2003, the Special-Interest Group for Information Technology Education (SIGITE) of the Association for Computing Machinery (ACM) established an IT Curriculum Writing Subcommittee to undertake the task of drafting a preliminary version of an IT volume for the CC2001 document. The charter of this committee was to take the material already created by the SIGITE Curriculum Committee and organize it into the form of the CS volume of the CC2001 document.

1.1 Overall structure of the CC2001 series

In light of the broadening scope of computing, it was decided to divide the CC2001 report into several volumes. This volume focuses specifically on Information Technology. To encompass the many other disciplines that are part of the overall scope of computing and Information Technology, however, IEEE-CS and ACM have created additional committees to undertake similar efforts in other areas, including Computer Engineering, Computer Science, Information Systems, and Software Engineering.

1.2 Overview of the process for developing this IT volume

The IT Curriculum Committee acknowledges the process used in developing the structure and content of the CC2001 document, and has made every effort to have the same structure in this document. The process is described in detail in the CS volume of the CC2001 document, and will therefore not be discussed in this volume.

Developing the recommendations in this volume is primarily the responsibility of the SIGITE Curriculum Committee-Writing Subcommittee, the members of which are listed at the beginning of this report. The CS (Computer Science) volume was the first volume completed of the CC2001 document, and has served as a model for the development of this 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, focuses on meeting the needs of users within an organizational and societal context through the **selection, creation, application, integration and administration** of computing technologies.

1.4 Broad goals of an IT program

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

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

1.5 Program outcomes

To enable IT graduates to achieve the above goals, they must possess the following skills upon graduation, namely the ability to:

- (a) Use and apply current technical concepts and practices in the core information technologies;
- (b) Analyze, identify and define the requirements that must be satisfied to address problems or opportunities faced by organizations or individuals;
- (c) Design effective and usable IT-based solutions and integrate them into the user environment
- (d) Assist in the creation of an effective project plan;
- (e) Identify and evaluate current and emerging technologies and assess their applicability to address the users' needs;
- (f) Analyze the impact of technology on individuals, organizations and society, including ethical, legal and policy issues;
- (g) Demonstrate an understanding of best practices and standards and their application;
- (h) Demonstrate independent critical thinking and problem solving skills;
- (i) Collaborate in teams to accomplish a common goal by integrating personal initiative and group cooperation;
- (j) Communicate effectively and efficiently with clients, users and peers both verbally and in writing, using appropriate terminology;
- (k) Recognize the need for continued learning throughout their career.

1.6 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 outlined above. It is intended as a guide for educational institutions of higher education in the creation and/or revision of baccalaureate programs in IT. It is expected that it will also be useful in the creation and/or revision of associate programs in IT.

This volume of the Computing Curricula report looks specifically at Information Technology. The main body of the report consists of 13 chapters. [Chapter 2](#) begins with a survey and analysis of past reports, focusing most closely on Computing Curricula 1991. [Chapter 3](#) outlines the changes that have recently occurred in the computing disciplines which gave rise to the IT discipline, and discusses the implications that those changes have for curriculum design and pedagogy. In [Chapter 4](#), we articulate a set of principles that have guided the development of this volume of the Computing Curricula report, as we have attempted to build on the strengths of our predecessors while avoiding some of the problems observed in earlier reports or volumes. [Chapters 5](#) and [6](#) present overviews of the Information Technology body of knowledge and the curriculum recommendations that are examined in detail in the appendices. [Chapter 7](#) describes the core courses and approaches we recommend for an IT program. Because these courses alone do not constitute a complete baccalaureate curriculum, [Chapter 8](#) summarizes additional courses and topics that must be included as part of the academic program. One important aspect of the complete curriculum involves the study of professional practice, which is discussed in [Chapter 9](#). In [Chapter 10](#), we outline a set of characteristics that define the successful Information Technology graduate. [Chapter 11](#) looks at the problem of teaching Information Technology and computing-related skills to students in other disciplines. Finally, [Chapter 12](#) offers strategic and tactical suggestions for addressing the institutional challenges that affect the implementation of this report.

The bulk of the material in this report appears in two appendices. [Appendix A](#) looks in detail at the body of knowledge for baccalaureate Information Technology programs. [Appendix B](#) consists of full descriptions for the recommended courses that comprise the sample curricula. We hope that providing both the body of knowledge and course descriptions helps departments to create effective curricula more easily than using either of these sources alone.



Computing Curricula Information Technology Volume

Chapter 2 Lessons from Past Reports

In developing this report, the IT Curriculum Committee did not have to start from scratch. We have benefited tremendously from past curriculum studies, and are indebted the authors of those studies and to the authors of the CS volume of the CC2001 report for their dedicated efforts. As part of our early work on this *Computing Curricula - IT Volume*, we looked carefully at the most recent curriculum studies -- particularly *Computing Curricula 2001* -- to get a sense of how those studies have influenced computing education. By identifying which aspects of the previous reports have been the most successful, we hoped to structure this volume to maximize its impact. This chapter offers an overview of the earlier reports and the lessons we have taken from them.

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 to be defined in the five volumes of this Computing Curricula report include Computer Science, Computer Engineering, Software Engineering, Information Systems, and Information Technology. We anticipate that there may be others in the future.

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

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

Since CITC-1, three more CITC conferences were held, and both the accreditation and curriculum committees worked on their assigned tasks. Many educational institutions sought input from members of their industrial advisory boards. In July of 2003, SITE became SIGITE, the Special Interest Group for Information Technology Education, of the ACM. Draft criteria for accreditation were approved by SITE and were posted for public approval in the Fall of 2003. The curriculum committee began working on this document in December of 2003, using as a model the CS volume of the CC2001 document.

2.2 Evaluation of curriculum efforts

It is felt by the members of SIGITE 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 goes out, it is hoped that continual input will be sought and incorporated to provide for a continually current Information Technology curriculum.



Computing Curricula Information Technology Volume

Chapter 3 Changes in the Information Technology 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 PC (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 Internet 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, Internet browsers became the first compelling reason for everyone in society to use a computer. The almost overnight acceptance of the WWW by society at large created a hyper-demand for Web-based content and services, which ignited the explosion in demand for Web content developers and Web masters. As Web sites became more active and interactive, the demand for application developers and especially database developers expanded as well.

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

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

3.1 Technical changes

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

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

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

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

3.2 Pedagogical changes

The technical changes that have driven the recent expansion of computing have direct implications on the culture of education. Computer networks, for example, make distance education much more feasible, leading to enormous growth in this area. Those networks also make it much easier to share curricular resources among widely distributed institutions. Technology also affects the nature of pedagogy. Demonstration software, computer projection, and individual laboratory stations have made

a significant difference in the way Information Technology is taught. The design of Information Technology curricula must take into account those changing technologies.



Computing Curricula Information Technology Volume

Chapter 4 Principles

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

1. *Although this document can in principle be used as a stand-alone document, the formulation of the curriculum was governed by the desire to provide a blueprint to create accreditable programs.* In particular, successful completion of the curriculum should enable a student to achieve the learning outcomes that were included in the IT Accreditation Criteria, namely,

Graduates of baccalaureate programs in Information Technology must have the ability to:

- (a) Use and apply current technical concepts and practices in the core information technologies;
- (b) Analyze, identify and define the requirements that must be satisfied to address problems or opportunities faced by organizations or individuals;
- (c) Design effective and usable IT-based solutions and integrate them into the user environment;
- (d) Assist in the creation of an effective project plan;
- (e) Identify and evaluate current and emerging technologies and assess their applicability to address the users' needs;
- (f) Analyze the impact of technology on individuals, organizations and society, including ethical, legal and policy issues;
- (g) Demonstrate an understanding of best practices and standards and their application;
- (h) Demonstrate independent critical thinking and problem solving skills;
- (i) Collaborate in teams to accomplish a common goal by integrating personal initiative and group cooperation;
- (j) Communicate effectively and efficiently with clients, users and peers both verbally and in writing, using appropriate terminology;
- (k) Recognize the need for continued learning throughout their career.

2. *This document is intended to exist as part of the CC2004 series.* The committee therefore followed the format developed in other documents within this series, in particular CC2001 – Computer Science, and adopted the terminology in that document to describe the IT body of knowledge. In particular, the IT body of knowledge is organized hierarchically into three levels. The highest level of the hierarchy is the **knowledge area**, which represents a particular disciplinary sub-field. The knowledge areas are broken down into smaller divisions called **units**, which represent individual thematic modules within

an area, and are defined in terms of a set of **topics** (the third level), and learning outcomes which help define these topics.

3. *Despite the rapidly evolving nature of Information Technology, the committee wanted to formulate a curriculum with some longevity.* In formulating knowledge areas for Information Technology, the committee 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, the committee recommends that the professional associations in Information Technology establish an ongoing review process that allows individual components of the curriculum recommendations to be updated on a recurring basis.

4. *The curriculum must be flexible and the required body of knowledge must be as small as possible.* There is 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 such 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, the committee recommends core outcomes that must be met, and provides example elective learning outcomes for additional depth in each unit of each knowledge area.

5. *The curriculum must reflect those aspects that set Information Technology apart from other computing disciplines.* The integration of different technologies and the integration of technologies into organizations are fundamental to Information Technology. An IT graduate must therefore acquire a skill set that enables him or her to successfully perform integrative tasks, including user advocacy skills, the ability to address information assurance and security concerns, the ability to manage complexity through abstraction, a deep understanding of the technology, adaptability, outstanding interpersonal skills, high ethical standards, and professional responsibility. The curriculum must reflect these pervasive themes, which are discussed further in Chapters 7, 8 and 10.

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

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

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

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

Computing Curricula Information Technology Volume



Chapter 5 Overview of the IT Body of Knowledge

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

ITF	Information Technology Fundamentals
HCI	Human Computer Interaction
IAS	Information Assurance and Security
IM	Information Management
IPT	Integrative Programming & Technologies
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 #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 thematic modules 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 elective 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, the SIGITE Curriculum Committee made a distinction within knowledge units between **core learning outcomes** and **elective learning outcomes**. Core learning outcomes describe skills for which there is a broad consensus that anyone obtaining a baccalaureate degree in the field must acquire. The model curriculum also specifies a set of elective learning outcomes, typically skills that must be acquired by students specializing in the knowledge area with which that unit is associated. While the core and elective outcomes associated with a knowledge unit typically cover the same topics, the depth to which the topic is covered and the skill levels that students are expected to achieve differ significantly between core and elective learning outcomes.

In discussing the IT2005 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 must 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 is not a complete curriculum.* Because the core is defined as minimal, it does not, by itself, constitute a complete baccalaureate curriculum.
- *The core must be supplemented by additional material.* Every baccalaureate program must include material relevant to additional elective learning outcomes from the body of knowledge. The IT2005 report does not define what those must be, as this additional work can and should vary based on institutional mission, the areas of concentration offered by a given institution, and individual student choice.
- *Core learning outcomes are not necessarily achieved in a set of introductory courses early in the baccalaureate 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, the curriculum committee believes that all students must create a significant system, including hardware and software installation and test, at some point during their baccalaureate program. The material that is essential to successful management of projects at this scale is therefore part of the core, since it is required of all students. At the same time, the project course experience is very likely to come toward the end of a student's baccalaureate program. Similarly, introductory courses may include material relevant to elective 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, the IT 2005 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:

- *The committee does not seek to endorse the lecture format.* Even though we have used a metric with its roots in a classical, lecture-oriented form, the committee believes, 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 will typically entail a total of 12 hours (3 in class and 9 outside).
- *The hours listed for a unit represent a minimum level of coverage.* The time measurements we have assigned for each unit should be interpreted as the *minimum* amount of time necessary to enable a student to achieve the learning outcomes for that unit. It is always appropriate to spend more time on a unit than the mandated minimum.

5.1.3 Packaging units into courses

The structure and format of courses vary significantly from institution to institution and from country to country. Even within the United States, some colleges and universities use a semester system while others follow a shorter quarter system. Under either system, there can be differences in the number of weeks in a semester, the number of lectures in a week, and the number of minutes in a lecture.

Appendix B includes a sample way 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, which units are core, and the minimum time required for each -- appears below. The details of the body of knowledge appear in [Appendix A](#).

The Information Technology Body of Knowledge

ITF. Information Technology Fundamentals (33 core hours)

- ITF1. Pervasive Themes in IT (17)
- ITF2. Organizational Issues (6)
- ITF3. History of IT (3)
- ITF4. IT and Its Related and Informing Disciplines (3)
- ITF5. Application Domains (2)
- ITF6. Applications of Math and Statistics to IT (2)

HCI. Human Computer Interaction (20 core hours)

- HCI1. Human Factors (6)
- HCI2. HCI Aspects of Application Domains (3)
- HCI3. Human-Centered Evaluation (3)
- HCI4. Developing Effective Interfaces (3)
- HCI5. Accessibility (2)
- HCI6. Emerging Technologies (2)
- HCI7. Human-Centered Software Development (1)

IAS. Information Assurance and Security (23 core hours)

- IAS1. Fundamental Aspects (3)
- IAS2. Security Mechanisms (Countermeasures) (5)
- IAS3. Operational Issues (3)
- IAS4. Policy (3)
- IAS5. Attacks (2)
- IAS6. Security Domains (2)
- IAS7. Forensics (1)
- IAS8. Information States (1)
- IAS9. Security Services (1)
- IAS10. Threat Analysis Model (1)
- IAS11. Vulnerabilities (1)

IM. Information Management (34 core hours)

- IM1. IM Concepts and Fundamentals (8)
- IM2. Database Query Languages (9)
- IM3. Data Organization Architecture (7)
- IM4. Data Modeling (6)
- IM5. Managing the Database Environment (3)
- IM6. Special-Purpose Databases (1)

IPT. Integrative Programming & Technologies (23 core hours)

- IPT1. Intersystems Communications (5)
- IPT2. Data Mapping and Exchange (4)
- IPT3. Integrative Coding (4)
- IPT4. Scripting Techniques (4)
- IPT5. Software Security Practices (4)
- IPT6. Miscellaneous Issues (1)
- IPT7. Overview of programming languages (1)

NET. Networking (20 core hours)

- NET1. Foundations of Networking (3)
- NET2. Routing and Switching (8)
- NET3. Physical Layer (6)
- NET4. Security (2)
- NET5. Application Areas (1)
- NET6. Network Management

PF. Programming Fundamentals (38 core hours)

- PF1. Fundamental Data Structures (10)
- PF2. Fundamental Programming Constructs (9)
- PF3. Object-Oriented Programming (9)
- PF4. Algorithms and Problem-Solving (6)
- PF5. Event-Driven Programming (3)
- PF6. Recursion (1)

PT. Platform Technologies (14 core hours)

- PT1. Operating Systems (10)
- PT2. Architecture and Organization (3)
- PT3. Computing infrastructures (1)
- PT4. Enterprise Deployment Software
- PT5. Firmware
- PT6. Hardware

SA. System Administration and Maintenance (11 core hours)

- SA1. Operating Systems (4)
- SA2. Applications (3)
- SA3. Administrative Activities (2)
- SA4. Administrative Domains (2)

SIA. System Integration and Architecture (21 core hours)

- SIA1. Requirements (6)
- SIA2. Acquisition/Sourcing (4)
- SIA3. Integration (3)
- SIA4. Project Management (3)
- SIA5. Testing and QA (3)
- SIA6. Organizational Context (1)
- SIA7. Architecture (1)

SP. Social and Professional Issues (23 core hours)

- SP1. Professional Communications (5)
- SP2. History of Computing (3)
- SP3. Social Context of Computing (3)
- SP4. Teamwork Concepts and Issues (3)
- SP5. Intellectual Properties (2)
- SP6. Legal Issues in Computing (2)
- SP7. Organizational Context (2)
- SP8. Professional and Ethical Issues and Responsibilities (2)
- SP9. Privacy and Civil Liberties (1)

WS. Web Systems and Technologies (21 core hours)

- WS1. Web Technologies (10)
- WS2. Information Architecture (4)
- WS3. Digital Media (3)
- WS4. Web Development (3)
- WS5. Vulnerabilities (1)
- WS6. Social Software

Total Hours: 281

Notes:

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

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

Computing Curricula Information Technology Volume



Chapter 6 Overview of the Curricular Models

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

6.1 Overall structure of the model curricula

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

It must be noted that the categories of introductory, intermediate and advanced are defined independent of core and elective 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 baccalaureate programs aim to provide their graduates with the skills and knowledge to take on appropriate professional positions in Information Technology on graduation and grow into leadership positions or pursue research or graduate studies in the field. IT professionals are primarily entrusted with the integration of different technologies, and the integration of the technology into organizations. This requires a familiarity with the technology that goes beyond the purely theoretical. IT baccalaureate 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 baccalaureate 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 five broad goals and the eleven program outcomes of IT programs include such words as *apply*, *employ*, *use*, *integrate*, *demonstrate*, and *accomplish*, all of which include a strong application component. While the

committee does 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,
- 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, the committee is 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

An IT curriculum may be implemented using one of several implementation strategies. The strategies addressed here are *practicum first*, *theory first*, *core/integration first* and *pillars first*. The practicum-first and theory-first strategies are orthogonal to the latter two; an IT curriculum may choose either or a hybrid of the first two strategies, along with either of the second two.

The practicum-first approach provides students with significant experiential learning early in the curriculum. This approach may be appropriate when developing a 2+2 curriculum or when internships are required early in an IT curriculum.

Though the committee is unaware of any baccalaureate programs that use the theory-first approach, this strategy may be appropriate to programs preparing a significant portion of their students for graduate school. In general, the committee is of the opinion that most programs would benefit from a hybrid of these two strategies.

The core/integration-first approach provides a comprehensive coverage of the core outcomes from the body of knowledge as well as the integration of these core outcomes early in the curriculum. An example of the core/integration-first strategy is given in [Appendix B](#). Finally, the pillars-first approach introduces the detail of the IT pillars first and provides the integration later in the curriculum.

6.4 Covering the core

Regardless of the pedagogical strategy used by an IT program, the knowledge areas and core outcomes defined in the body of knowledge must be covered. Table 2 of [Appendix B](#) shows how the core outcomes would be satisfied by the curricular example given.



Computing Curricula Information Technology Volume

Chapter 7 The Core in the Curriculum

This chapter looks at the introductory phase of the baccalaureate curriculum, when students receive their first college-level exposure to computing classes. Section 7.1 outlines our overall philosophy concerning the introductory curriculum. Section 7.2 looks at the topics that are central to the design of introductory courses. Section 7.3 addresses the intermediate level, and section 7.4 describes the topics addressed at the advanced level.

7.1 Overall philosophy

With the proliferation of the Internet, and particularly the World-Wide Web, the field of Information Technology has grown very rapidly. Likewise, the academic discipline of Information Technology has developed quickly at many institutions, and its curriculum is now being formalized. As with any new discipline, there are many institutional implementations, each with its respective advantages and disadvantages. However, it is with much accord that we present this proposed curriculum with the full backing of the SIGITE Curriculum Committee. While we readily acknowledge it is not the only way to implement an IT program at a 4-year educational institution, we do believe it to be one of the best.

Information Technology is very much an integrative discipline; it pulls together the IT pillars of databases, human-computer interaction, networking, programming, and web systems and uses a solid background in each of them to enable graduates to solve all types of computing and informational problems, regardless of their origin. As a discipline, IT emphasizes 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.

As discussed previously in [6.3](#), there are four main implementation strategies presently envisioned: practicum first, theory first, core/integration first and pillars first. One of these strategies (core/integration first) is described in detail in [Appendix B](#).

7.2 Core Material to be Covered by the End of the second year (Introductory)

The core material we believe should be covered by the end of the second year of a 4-year IT program includes: 1) IT Fundamentals; 2) other introductory material.

7.2.1 IT Fundamentals

We believe that the introductory sequence is best covered in the first year of an IT program. One of the key pieces of this first year is the IT fundamentals, covering topics such as 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.

Throughout the deliberations of the SIGITE Curriculum Committee, 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
- a deep understanding 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 sequence in Appendix B illustrates how this could be achieved by indicating the courses in which these themes would be addressed.

7.2.2 Other Introductory Material

The other material to be covered at the IT introductory level includes Programming fundamentals (an IT knowledge area), and Computer architecture and Operating systems (part of the Platform Technologies knowledge area).

These knowledge areas for IT students may not be significantly different from those with similar titles required for students in computer science, software engineering, computer engineering, or information systems. Therefore, we do not feel it is essential that these materials be taught by IT faculty. Indeed, an institution can structure and teach these knowledge areas such that they may be taken by students in any of the computing disciplines. This would have the advantage that, if the student decided to choose another computing major, these courses would still count toward their new major. Additional details about how these may be structured into courses may be found in [Appendix B](#) – Course Descriptions.

7.3 Core Material to be Covered by the End of the third year (Intermediate)

The core material we believe should be covered by the end of the third year of a 4-year IT program includes the remaining 4 pillars of the IT curriculum (networking, web systems, information management, and human-computer interaction), system administration, system integration and architecture, and integrative programming.

7.3.1 Remaining IT Pillars

The five pillars of an IT curriculum are programming, networking, web systems, information management, and human-computer interaction. We have recommended above that programming be covered in the introductory material; the remaining four pillars (also knowledge areas) should be covered in the intermediate material. It is the feeling of the committee that these four knowledge areas are best studied after students have been introduced to them briefly in the introductory material, and after the students have learned the basics of programming in an appropriate high-level language. These knowledge areas may be covered as individual courses, or as modules in a more flexible course structure.

7.3.2 System Integration and Architecture and System Administration

The committee feels that these materials are best suited to the intermediate level, due to the more integrative and higher-level nature of the material. Both of these topics are knowledge areas in the IT curriculum, and may be covered in separate courses or in a single course combining the material. Additionally, this material could be covered as part of other courses if the structure is appropriate.

7.3.3 Integrative Programming

This IT knowledge area is also deemed appropriate for the intermediate level because of the need for the students to have already covered IT fundamentals, and to have had an introduction to programming. Due to the integrative nature of this knowledge area, it may also be helpful for the students to have already been introduced to the material in the platform technologies and web systems knowledge areas.

7.4 Core Material to be Covered by the End of the fourth year (Advanced)

The core material we believe should be covered by the end of the fourth year of a 4-year IT program includes: 1) information assurance and security, 2) the social and professional issues of IT, 3) an integrative capstone experience, 4) other advanced materials necessary to fill out the IT curriculum, and 5) the IT pervasive themes.

7.4.1 Information Assurance and Security (IAS)

The committee believes that this material is best covered after the students have been fully exposed to the five pillars of an IT curriculum, as well as all the other material at the introductory and intermediate levels. IAS is a very integrative knowledge area, and it is very difficult to teach the concepts and issues of IAS without a significant background in Information Technology.

7.4.2 Social and Professional Issues of IT

The social and professional issues of IT are not significantly different from those in other computing disciplines, with the possible exception of their intensification. IT graduates will be required to work closely with professionals from all computing disciplines, as well as people from all backgrounds. IT graduates will be qualified to work anywhere there is an IT infrastructure, whether in health care, government, law enforcement, biology, or an industrial setting. IT professionals must therefore be particularly adept at working with people of many backgrounds and perspectives. They must also have strong ethical convictions and accept responsibility for professional and personal behavior.

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

7.4.3 The Integrative Capstone Experience

The concept of a capstone type of experience in the last part of the curriculum has gained wide support in academia, particularly in the engineering and engineering technology disciplines, and is gaining support in the computing disciplines. The three common elements of nearly all capstone programs are: 1) students are divided into teams of typically 4 to 8 students each; 2) each team is given a real world project or problem to solve; 3) this project takes many weeks to complete (typically 14 or more). There is much research to support the effectiveness of this experience (see Jones SA, Houghtalen R; Dutta D, et al; Butkus, M. A., and Kelley, M.B; Devon, R., et al.; Catalano, G.D.; and Bohmann, L.J., et al. among many others). Due to the highly integrative nature of the experience, and the need for the students to have covered most of the IT curriculum prior to this experience, this material is best covered as close as possible to the end of the fourth year.

7.4.4 Other Advanced Materials

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

7.4.4.1 Optional IT Classes

The core by itself constitutes approximately 280 hours of instruction, which in a 4-year program of approximately 1,800 hours of instruction, constitutes only about 16%. For optional IT classes, we recommend courses that cover the core material in greater depth, and courses that cover the IT field in greater breadth, as deemed appropriate to each institution, so that the IT major is as complete as all other majors at a given institution.

7.4.4.2 Emphasis Classes

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

7.4.5 IT Pervasive Themes

The pervasive themes of the IT curriculum were previously described in section 7.2.1 under the IT fundamentals. They are mentioned again here, because they must be fully covered by the completion of the 4-year curriculum. They must be mentioned often, from the first year through the fourth, since they do not readily fit into a single course.



Computing Curricula Information Technology Volume

Chapter 8 Completing the Curriculum

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

8.1 General requirements

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

8.1.1 Mathematics

Mathematics techniques and formal mathematical reasoning are important in most areas of Information Technology. Mathematics provides a language for working with ideas relevant to the field of IT, specific tools for analysis and verification, and a theoretical framework for understanding important computing ideas. It is also important for IT programs to include enough mathematics so that students may converse with other computing professionals.

Given the important role of mathematics within Information Technology, the IT curriculum must include an appropriate set of mathematics. Basic mathematical concepts should be introduced early within a student's course work, and later courses should use these concepts. While different institutions will need to adjust their mathematics requirements to reflect local needs and opportunities, it is important that the IT curriculum include at least the mathematics described in this document.

The SIGITE Curriculum Committee makes the following recommendations with respect to the mathematical content of the IT curriculum:

- *Discrete mathematics.* All students need exposure to the tools of discrete mathematics. While additional courses are available, it is essential that IT students to receive adequate exposure in this area.
- *Statistics.* Students should take at least one course in statistics so that they can be familiar with the terms and methods of statistical analysis. We also believe they would benefit greatly from learning about statistical probability.

8.1.2 The scientific method

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

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

The SIGITE Curriculum Committee therefore makes the following recommendations about science:

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

8.1.3 Familiarity with application domains

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

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

8.1.4 Communications skills

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

- Communicate ideas effectively in written form
- Make effective oral presentations, both formally and informally
- Understand and offer constructive critiques of the presentations of others
- Have a pleasant demeanor as they work with people on their IT needs, either in person or by phone
- Write appropriate emails 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.

8.1.5 Working in teams

Few computing professionals can expect to work in isolation for very much of the time. Information technology projects are usually implemented by groups of people working together as a team. Information technology students therefore need to learn about the mechanics and dynamics of effective team participation as part of their baccalaureate education. Moreover, because the value of working in teams (as well as the difficulties that arise) does not become evident in small-scale projects, students need to engage in team-oriented projects that extend over a reasonably long period of time, possibly a full semester or a significant fraction thereof.

To ensure that students have the opportunity to acquire these skills as undergraduates, the SIGITE Curriculum Committee recommends that all IT programs include the following:

- Opportunities to work in teams beginning relatively early in the curriculum.
- A significant project that involves a complex implementation task in which both the design and implementation are undertaken by a small student team. This project is often scheduled for the last year of undergraduate study, where it can serve as a capstone for the undergraduate experience. Some of the details of this experience were discussed previously in [Chapter 8](#).

The experience that students derive from a significant team project can be enhanced further by using teams that cross disciplinary boundaries. As an example, IT students can be paired with students in biology to conduct a project in the emerging area of biocomputation. Such a project will require expertise from both disciplines, along with strategies to support effective communication across the disciplinary boundary. The

ABET 2000 report specifically endorses the concept of interdisciplinary team projects, and the SIGITE Curriculum Committee agrees that such projects can provide a rich and valuable experience for students, both inside and outside of Information Technology.

8.1.6 Becoming a Well-Rounded 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 and the diverse opinions and perspectives of others, keeping abreast of social and political developments, and contributing in a positive way to their neighborhood and community.

8.1.7 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 material of the main curriculum. At the same time, these pervasive themes are considered essential for IT students, and must be adequately taught. There are delicate issues of balance here, and curriculum and course designers must find the proper mix.

8.2 Advanced courses

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

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

Figure 8-1. Examples of advanced courses by area

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



Computing Curricula Information Technology Volume

Chapter 9 Professional Practice

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

9.1 Rationale

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

Both the private and public sectors have a vested interest in students learning professional practice. They find that students who have experience with the realities of professional work understand the value of interpersonal skills in collaborating with team members and clients, maintain their focus on producing high-quality work, adhere to strong ethical convictions, contribute their time and talents to worthy outside causes, engage in life-long learning, and participate in improvements in their firm. Each year, for example, the National Association of Colleges and Employers conducts a survey to determine what qualities employers consider most important in applicants seeking employment [[NACE2001](#)]. In 2001, the top ten factors were

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 are 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 [[ACM2001](#), [IEEE2001](#), [SEEPP98](#)] 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 [[ABET2000](#), [CSAB2000](#)].
- 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 baccalaureate 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. One of 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 that would meet CSAB Criteria 2000 [[CSAB2000](#)]. Although many schools clearly did not consider this material to be essential, it is encouraging that more than half of the schools did. With the growing emphasis on professionalism in accreditation criteria, it is likely that other schools will strengthen their commitment to teaching professional practice.

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

- *Senior capstone courses.* These courses typically form a one- to a two-semester sequence during the student's last year. Usually, students must work in teams to design and implement projects, where those projects must involve consideration of real-world issues including cost, safety, efficiency, and suitability for the intended user. The projects may be developed solely for the class, but may also involve other on- or off-campus clients. Although the emphasis of the course is on project work and student presentations, some material on intellectual property rights, copyrights, patents, law, and ethics may be included.
- *Professionalism, ethics, and law courses.* These courses are one semester long and expose students to issues of professional practice, ethical behavior, and computer law. Topics included may be history of computing, impact of computers on society, computing careers, legal and ethical responsibilities, and the computing profession.
- *Practicum/internship/co-op programs.* These programs are sponsored by the institution (preferably) or department to allow students to have the opportunity to work in industry full- or part-time before graduation. At least one to two coordinators should be hired to oversee the program and it is helpful to have one coordinator at the college level as well as to have a part-time coordinator within a department. Students typically work during the summers and/or from one to three semesters while they are engaged in their baccalaureate 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 web 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 for 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, private/public sectors and 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 professional society ethical and professional behavior guidelines. 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, techniques for tackling ill-defined problems, teamwork with individual accountability, real-life ethics issues, standards and guidelines, 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 problematical 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. As noted at the end of section 10.2, these courses may be from disciplines outside of Information Technology and still have a profound effect on the professional development of students.

9.5 Assessing professional practice work

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

- Outcomes-based assessment
- Reviewing assignments, projects, and exams for appropriate inclusion of professional practice material
- Critically reviewing and establishing sound measurements on student work to show student progress and improvement
- Getting students involved in the review and assessment process so that they get a better sense of the assessment process

- Employing professionals in the private and public sectors to help in assessing student project work
- Using standardized tests to measure overall student progress
- Taking post-graduation surveys of alumni to see how well alumni thought their education prepared them for their careers
- Obtaining accreditation to demonstrate that certain education standards for professional practice have been met

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

9.6 Certifications

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

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



Computing Curricula Information Technology Volume

Chapter 10 Characteristics of IT Graduates

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

- (a) Use and apply current technical concepts and practices in the core information technologies;
- (b) Analyze, identify and define the requirements that must be satisfied to address problems or opportunities faced by organizations or individuals;
- (c) Design effective and usable IT-based solutions and integrate them into the user environment;
- (d) Assist in the creation of an effective project plan;
- (e) Identify and evaluate current and emerging technologies and assess their applicability to address the users' needs;
- (f) Analyze the impact of technology on individuals, organizations and society, including ethical, legal and policy issues;
- (g) Demonstrate an understanding of best practices and standards and their application;
- (h) Demonstrate independent critical thinking and problem solving skills;
- (i) Collaborate in teams to accomplish a common goal by integrating personal initiative and group cooperation;
- (j) Communicate effectively and efficiently with clients, users and peers both verbally and in writing, using appropriate terminology;
- (k) Recognize the need for continued learning throughout their career.

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

User centeredness and advocacy. IT graduates do not 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, as well as a realization that solutions to problems that have arisen 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, 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 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.

A deep understanding of 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 a deep technical expertise 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 baccalaureate 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, 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 used by users into technical language, and vice versa. This in turn requires an appreciation of adjacent organizational functions, and an awareness of organizational culture. It also requires respect for and appreciation of diversity and the ability to tolerate and appreciate different points of views and approaches to problems or opportunities that arise.



Computing Curricula Information Technology Volume

Chapter 11 Computing Across Curriculum

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

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

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

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

11.1 Process questions

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

11.1.1 Course specification

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

There are four possible goals of a general-education course in Information Technology:

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

The first step in developing new general-education courses is identifying a curricular need that is not currently being met. This may be done either reactively or proactively. Information technology departments should certainly respond to requests from faculty or industry representatives for a new course that could be quite useful to their students or employees. Alternately, Information Technology 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

Information Technology 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.
- *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 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 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.
- *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.



Computing Curricula Information Technology Volume

Chapter 12 Institutional Challenges

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

12.1 The need for local adaptation

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

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

Creating a workable curriculum requires finding an appropriate balance among these factors, which will require different choices at every institution. There can be no single curriculum that works for all institutions.

Every college and university will need to consider the various models proposed in this document and design an implementation that meets the need of that environment.

12.2 Principles for curriculum design

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

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

12.3 The need for adequate computing resources

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

At the same time, it is essential for institutions to recognize that computing 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, particularly if one includes the development costs of courseware. Providing adequate support staff to maintain the computing facilities represents another large expense. To be successful, Information Technology programs must receive adequate funding to support the computing needs of both faculty and students.

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

12.4 Attracting and retaining faculty

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

- *Adopt an aggressive plan for faculty recruitment.* Scarcity is in itself no reason to abandon the search; the shortage of candidates simply means that Information Technology departments need to look harder. Being successful is usually a matter of initiative and persistence. Departments must start the recruiting process very early and should consider reaching out to a wide range of potential applicants, including overseas students and people currently working in industry.
- *Create academic positions that focus on teaching.* As in most disciplines, faculty positions in Information Technology typically require a Ph.D. and involve both research and teaching. If there were a sufficient pool of candidates with the right credentials and skills, insisting on these qualifications would cause no problem. Given the shortage of faculty candidates, it is not clear whether Information Technology departments can afford such selectivity. It is not necessary for every institution to maintain a research program in Information Technology. At the same time, it is hard to imagine that any university today could get away without offering courses in this area. Opening faculty positions to those who enjoy teaching but are not drawn to academic research increases the size of the available pool.
- *Make sure that faculty receive the support they need to stay in academia.* Studies undertaken by the National Science Foundation in the 1980s found that faculty members who left academia for industry typically did not cite economics as their primary motivation [Curtis83]. Instead, they identified a range of concerns about the academic work environment, huge class sizes, heavy teaching loads, inadequate research support, the uncertainty of tenure, and bureaucratic hassles, that the NSF study refers to collectively as "institutional disincentives." As enrollments in Information Technology courses rise, it is critical for institutions to ensure that faculty workloads remain manageable.

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

12.5 Conclusion

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

Bibliography

Beaulieu, M. *Wireless Internet application and architecture*. Boston, MA: Addison-Wesley (2002).

Bohmann, L.J, Mork, B.A., Wiitanen, D.O.; “Power engineering design projects: Capstone team projects versus topical design courses”; *IEEE Transactions on Power Systems*; Feb 2004; 19:1, pp 152-156.

Butkus, M. A., and Kelley, M. B.; “Approach for integrating professional practice issues into undergraduate environmental engineering design projects”; *Journal of Professional Issues in Engineering Education and Practice*; Jul 2004; 130:3, pp 166-172.

Catalano, G.D.; “Senior capstone design and ethics: A bridge to the professional world”; *Science and Engineering Ethics*; Apr 2004; 10:2, pp 409-415.

Chumer, M.J.; “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; 2002.

Curtis, Kent K. “Computer manpower: Is there a crisis?” Washington DC: National Science Foundation, 1983. <http://www.acm.org/sigcse/papers/curtis83/>.

Devon, R., Bilen, S., McKay, A., De Pennington, A., Serrafiero, P., and Sierra, J.S.; “Integrated design: What knowledge is of most worth in engineering design education?”; *International Journal of Engineering Education*; 2004; 20:3, pp 424-432.

Dutta D., Geister D.E., Tryggvason G.; “Introducing hands-on experiences in design and manufacturing education”; *International Journal of Engineering Education*; 2004; 20:5, pp 754-763.

Jones SA, Houghtalen R; “Using senior design capstone as model for graduate education”; *Journal of Professional Issues in Engineering Education and Practice*; Apr 2000; 126:2, pp 83-88

Maconachy, Schou, Ragsdale, Welch: A Model for Information Assurance: An Integrated Approach; Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, US Military Academy, West Point, NY 5-6 June 2001.

Roberts, Eric, Lilly, John, and Rollins, Bryan. “Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience.” *SIGCSE Bulletin* (27)1:48-52, March 1995.

Index

Accreditation 2.1, 4.1, 9.1, 9.2, 9.3.3, 9.5, 12.2

ACM Chapter 1 intro, 1.1, 2.1, 9.1, 9.2

Adaptability 4.1, 7.2.1, 9.1, 10.1

Application domains 3.1, 5.2, 8.1.3, chpt 11 intro

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

CC2001 1 intro, 1.1, 1.2, 2 intro, 2.1, 4.1, 5.1.2, 11 intro, 12 intro

CITC 2.1

Communication skills 8.1, 8.1.4, 9.1, 9.3.2, 11 intro, 11.1.3

Complexity 7.1, 7.2.1, 10 intro, 11 intro

Core table of contents, 1.5, 1.6, 4.1, 5.1., 5.1.2, 5.2, 6.1, 6.3, 6.4, chpt 7 intro, 7.1, 7.2, 7.3, 7.4, 7.4.4, 8 intro, 8.2, 9.1, 10 intro

Databases 3.1, 4.1, 5.2, 7.1, 7.3, 7.3.1, 8.2, 10.1

Distance education 3.2

E-commerce 3.1, 8.2

Elective 4.1, 5.1.1, 6.1, 7.4.4, 11.1.1

Ethics 7.2.1, 8.2, 9.1, 9.2, 9.4, 11 intro

Experiential learning 6.2, 6.3

General education 9.2, 11.1.1

Graduate programs 3 intro, 6.3, 12 intro, 12.1

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

IEEE Computer Society

Implementation strategy

Industrial advisory board (IAB) 2.1, 9.3.1, 9.3.3

Internet 3 intro, 3.1, 7.1, 9.4

Interpersonal skills 4.1, 7.2.1, 9.1, 9.3.2, 10.1

IT – Definition 1.3

IT – Broad goals 1.4

IT – Program outcomes 1.5

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

Learning outcomes 4.1, 6.2

Life-long learning 7.2.1, 9.1, 10.1

Mathematics 2.1, 8.1.1, 9.2, 11.1.1, 11.2

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

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

Pillars 6.3, 7.1, 7.3, 7.3.1, 7.4.1

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

Probability 8.1.1

Problem solving 1.5, 4.1, 10.1, 11 intro

Professionalism 7.2.1, 9.2, 12.2

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

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

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

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

SITE 2.1

Statistics 5.2, 8.1.1

Teamwork 5.2, 9.1, 9.2, 9.3.2, 9.4

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

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

World-Wide Web (WWW) 3.1, 7.1

Curriculum Committee meetings:

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



Appendix A

The IT Body of Knowledge

This appendix to the Computing Curricula – Information Technology Volume report defines the knowledge domain that is likely to be taught in a baccalaureate 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, the curriculum committee feels 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 elective units

The curriculum committee has 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 has at one time been considered fundamental to any particular computing discipline. The curriculum committee has therefore sought to define a minimal **core** consisting of that material that essentially everyone teaching Information Technology agrees is essential to anyone obtaining a baccalaureate degree in this field. Material offered as part of a baccalaureate program that falls outside the core is considered to be **elective**. By insisting on a broad consensus in the definition of the core, the task force hopes to keep the core as small as possible, giving institutions the freedom to tailor the elective 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 is not a complete curriculum.* Because the core is defined as minimal, it does not, by itself, constitute a complete baccalaureate curriculum. Every baccalaureate program must include additional elective units from the body of knowledge, although this report does not define what those units will be.

- *Core units are not necessarily limited to a set of introductory courses taken early in the baccalaureate curriculum.* Although many of the units defined as core are indeed introductory, there are also some core units that clearly must be covered only after students have developed significant background in the field. For example, the curriculum committee believes that all students must develop a substantial project at some point during their baccalaureate program. The material that is essential to successful management of projects at this scale is therefore part of the core, since it is required of all students. At the same time, the project course experience is very likely to come toward the end of a student's baccalaureate program. Similarly, introductory courses may include elective units alongside the coverage of core material. The designation *core* simply means *required* and says nothing about the level of the course in which it appears.

Assessing the time required to cover a unit

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

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

Details of the IT body of knowledge

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

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

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

The Information Technology Body of Knowledge

ITF. Information Technology Fundamentals (33 core hours)

- ITF1. Pervasive Themes in IT (17)
- ITF2. Organizational Issues (6)
- ITF3. History of IT (3)
- ITF4. IT and Its Related and Informing Disciplines (3)
- ITF5. Application Domains (2)
- ITF6. Applications of Math and Statistics to IT (2)

HCI. Human Computer Interaction (20 core hours)

- HCI1. Human Factors (6)
- HCI2. HCI Aspects of Application Domains (3)
- HCI3. Human-Centered Evaluation (3)
- HCI4. Developing Effective Interfaces (3)
- HCI5. Accessibility (2)
- HCI6. Emerging Technologies (2)
- HCI7. Human-Centered Software (1)

IAS. Information Assurance and Security (23 core hours)

- IAS1. Fundamental Aspects (3)
- IAS2. Security Mechanisms (Countermeasures) (5)
- IAS3. Operational Issues (3)
- IAS4. Policy (3)
- IAS5. Attacks (2)
- IAS6. Security Domains (2)
- IAS7. Forensics (1)
- IAS8. Information States (1)
- IAS9. Security Services (1)
- IAS10. Threat Analysis Model (1)
- IAS11. Vulnerabilities (1)

IM. Information Management (34 core hours)

- IM1. IM Concepts and Fundamentals (8)
- IM2. Database Query Languages (9)
- IM3. Data Organization Architecture (7)
- IM4. Data Modeling (6)
- IM5. Managing the Database Environment (3)
- IM6. Special-Purpose Databases (1)

IPT. Integrative Programming & Technologies (23 core hours)

- IPT1. Intersystems Communications (5)
- IPT2. Data Mapping and Exchange (4)
- IPT3. Integrative Coding (4)
- IPT4. Scripting Techniques (4)
- IPT5. Software Security Practices (4)
- IPT6. Miscellaneous Issues (1)
- IPT7. Overview of programming languages (1)

NET. Networking (20 core hours)

- NET1. Foundations of Networking (3)
- NET2. Routing and Switching (8)
- NET3. Physical Layer (6)
- NET4. Security (2)
- NET5. Application Areas (1)
- NET6. Network Management

PF. Programming Fundamentals (38 core hours)

- PF1. Fundamental Data Structures (10)
- PF2. Fundamental Programming Constructs (9)
- PF3. Object-Oriented Programming (9)
- PF4. Algorithms and Problem-Solving (6)
- PF5. Event-Driven Programming (3)
- PF6. Recursion (1)

PT. Platform Technologies (14 core hours)

- PT1. Operating Systems (10)
- PT2. Architecture and Organization (3)
- PT3. Computing infrastructures (1)
- PT4. Enterprise Deployment Software
- PT5. Firmware
- PT6. Hardware

SA. System Administration and Maintenance (11 core hours)

- SA1. Operating Systems (4)
- SA2. Applications (3)
- SA3. Administrative Activities (2)
- SA4. Administrative Domains (2)

SIA. System Integration and Architecture (21 core hours)

- SIA1. Requirements (6)
- SIA2. Acquisition/Sourcing (4)
- SIA3. Integration (3)
- SIA4. Project Management (3)
- SIA5. Testing and QA (3)
- SIA6. Organizational Context (1)
- SIA7. Architecture (1)

SP. Social and Professional Issues (23 core hours)

- SP1. Professional Communications (5)
- SP2. History of Computing (3)
- SP3. Social Context of Computing (3)
- SP4. Teamwork Concepts and Issues (3)
- SP5. Intellectual Properties (2)
- SP6. Legal Issues in Computing (2)
- SP7. Organizational Context (2)
- SP8. Professional and Ethical Issues and Responsibilities (2)
- SP9. Privacy and Civil Liberties (1)

WS. Web Systems and Technologies (21 core hours)

- WS1. Web Technologies (10)
- WS2. Information Architecture (4)
- WS3. Digital Media (3)
- WS4. Web Development (3)
- WS5. Vulnerabilities (1)
- WS6. Social Software

Total Hours: 281

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.

IT Fundamentals (ITF) – 33 core hours

ITF1. Pervasive Themes in IT

ITF2. Organizational Issues

ITF3. History of Information Technology

ITF4. IT and Its Related and Informing Disciplines

ITF5. Application Domains

ITF6. Application of Math and Statistics to IT

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.

ITF1. Pervasive Themes in IT

Minimum core coverage time: 17 hours

Topics:

User centeredness and advocacy

Information assurance and security

IT systems model

Management of Complexity (abstraction, modeling, best practices, patterns, standards, and use of appropriate tools)

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 role of the IT professional as the user advocate.
6. Explain why life-long learning and continued professional development is critical for an IT professional.
7. Explain why adaptability and interpersonal skills are important to an IT professional.
8. Distinguish between data and information, and describe the interrelationship.
9. Describe the importance of data and information in IT.
10. Explain why the mastery of information and communication technologies is important to an IT professional.
11. Explain why the IAS perspective needs to pervade all aspects of IT.

ITF2. Organizational Issues

Minimum core coverage time: 6 hours

Topics:

- How to introduce IT applications
- Dissemination of innovation and change agency
- Integration of processes
- Business process redesign
- Cost benefit analysis
- Project Management

Core learning outcomes:

1. Describe the elements of a feasible IT application
2. Identify the extend and activities involved in an IT application
3. Understand the requirements of the business processes
4. Outline the project management processes
5. List the integration processes

ITF3. 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.
2. Describe significant impacts of computing on society.
3. Describe significant changes in human-computer interaction.
4. Outline the history of the Internet.

ITF4. IT and Its Related and Informing Disciplines

Minimum core coverage time: 3 hours

Topics:

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

Core learning outcomes:

1. Define “Information Technology”.
2. Describe the relationship between IT and other computing disciplines.
3. Describe the relationship between IT and non-computing disciplines.
4. Explain why mathematics and statistics are important in IT.

ITF5. 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
(and others)

Core learning outcomes:

1. Describe the application of IT in non-computing disciplines.
2. Describe how IT has impacted almost all aspects of modern living.
3. Describe ways and extends in which IT has changed the interaction and communication in our society.
4. Describe how IT has impacted the globalization of world economy, culture, political systems, health, security, warfare, etc.

ITF6. Application of Math and Statistics to IT

Minimum core coverage time: 2 hours

Topics:

Functions, relations, and sets
Data representations and encryptions
Basic logic used in programming
Applications of probability tools in solving problems
Applications of statistical methods in solving problems

Core learning outcomes:

1. Recognize the foundation of IT is built upon the various aspects of mathematics
2. Understand the number systems used in computation
3. Explain data representation and encoding systems
4. Describe the current encryption methods and their limitations
5. Describe the pervasive usage of mathematical concepts, such as functions, relations, sets as well as basic logic used in programming
6. Recognize the value of probability and statistics in projection
7. Describe the basic data analysis concepts and methods used in IT applications

Human-Computer Interaction (HCI) – 20 core hours

HCI 1. Human Factors

HCI 2. HCI Aspects of Application Domains

HCI 3. Human-Centered Evaluation

HCI 4. Developing Effective Interfaces

HCI 5. Accessibility

HCI 6. Emerging Technologies

HCI 7. Human-Centered Software

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: 6 hours

Topics:

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

Understanding the user

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

Ergonomics

Core learning outcomes

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

Elective learning outcomes:

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

HCI2. HCI Aspects of Application Domains

Minimum core coverage time: 3 hours

Topics:

Types of environments

Cognitive models

Approaches

Core learning outcomes

1. Describe several constraints on developing a user interface on a web page not encountered in developing a GUI for a standalone application.
2. Describe several affordances of a web environment that can enhance the usability of a web-based application.
3. Describe the connection between the development of a user interface and the user's knowledge of an application domain.
4. Describe different types of interactive environments other than a computer with a GUI.
5. Match descriptions of cognitive models with the names of the model.

Elective learning outcomes:

1. Develop user interfaces for domain specific applications
2. Interview domain experts to capture domain specific requirements for user interfaces.
3. Apply cognitive models to the design of application user interfaces.

HCI3. Human-Centered Evaluation

Minimum core coverage time: 3 hours

Topics:

Heuristics
Usability testing
Usability standards – international, OS, accessibility

Core learning outcomes

1. Describe the components used in a heuristic evaluation.
2. Demonstrate the steps necessary to perform a simple usability test for an existing software application.
3. Classify usability performance and preference metrics: learning, task time and completion, satisfaction.
4. Describe the major usability guidelines and standards.

Elective learning outcomes:

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

HCI4. Developing Effective Interfaces

Minimum core coverage time: 3 hours

Topics:

Understanding the user experience
Understanding interaction styles
Matching interface elements to user requirements
Graphical user interfaces
Non-graphical user interfaces
Localization, globalization

Development Tools - Environments, html Prototyping

Core learning outcomes

1. Describe ways in which users' characteristics (i.e., age, education, cultural differences, etc.) require adaptation of a user interface to increase effectiveness.
2. Define the different types of interaction styles.
3. Describe options for user interfaces for devices that do not support visual output capable of presenting a GUI.
4. Enumerate the options (techniques) for developing prototypes of user interfaces.
5. Describe the differences between low-fidelity and high-fidelity prototyping.
6. List examples of localization and globalization that would impact design.

Elective learning outcomes:

1. Select an appropriate interaction style for a task.
2. Design a low-fidelity GUI interface for an application which can be evaluated to guide development for a higher-fidelity prototype.
3. Using a prototyping tool, develop a high-fidelity prototype of a GUI which incorporates feedback from early usability testing.
4. Develop prototype examples that use different I/O modalities to accomplish to same task.
5. Develop prototype interfaces for devices that do not have a graphical display.
6. Given examples of the type of input required from a user, rank input options for interaction style based on expected effectiveness.

HCI 5. Accessibility

Minimum core coverage time: 2 hours

Topics:

Biometrics
Repetitive stress syndrome
ADA 508

Core learning outcomes

1. List some of the advantages and disadvantages of limiting access via biometrics.
2. Describe the symptoms of repetitive stress syndrome and list some of the approaches that can ameliorate the problem.
3. List some of the impacts on IT of the ADA 508 law.

Elective learning outcomes:

1. Choose a biometric access system for a given application and include the criteria used for making the choice.
2. Describe some of the changes that have yet to be made to computers to accommodate disabled users.

HCI 6. Emerging Technologies

Minimum core coverage time: 2 hours

Topics:

Alternative input/output devices
Alternative displays (heads-up, goggles, etc.)

Mobile computing
Wearable computing
Ubiquitous computing
Pervasive computing
Sensor-nets

Core learning outcomes

1. List several of the emerging alternative I/O devices for computers.
2. List several display types, especially for portable applications.
3. Describe the difference between mobile computing and wearable computing.
4. Give examples of ubiquitous computing, and compare them to pervasive computing examples.
5. List some of the advantages of sensor nets.

Elective 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.

HCI 7. Human-Centered Software

Minimum core coverage time: 1 hour

Topics:

User-centered design methods
Software development lifecycles
User analysis – Profiles, personas
Task analysis
 Scenarios
 Use Cases

Core learning outcomes

1. Explain the characteristics of human-centered design methods.
2. Describe the different software development lifecycles.
3. Describe the attributes of usability.
4. List the advantages and disadvantages for using a human-centered software development approach.
5. Analyze a user population and develop generalized profiles for each.
6. Compare user groups and develop appropriate personas to represent them.
7. Survey the user environment for appropriate tasks.
8. Describe in scenario form a problem situation to be addressed by a new or redesigned product.

Elective learning outcomes:

1. Integrate appropriate user-centered design methodologies into the development of an application or product.
2. Select appropriate user personas to be used during the product design phase.
3. Select appropriate user tasks for the application under consideration.
4. Measure the effectiveness of a design of an application or product during different stages of a software development lifecycle.

Information Assurance and Security (IAS) – 23 core hours

IAS1. Fundamental Aspects

IAS2. Security Mechanisms (Countermeasures)

IAS3. Operational Issues

IAS4. Policy

IAS5. Attacks

IAS6. Security Domains

IAS7. Forensics

IAS8. Information States

IAS9. Security Services

IAS10. Threat Analysis Model

IAS11. Vulnerabilities

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

IAS1. Fundamental Aspects

Minimum core coverage time: 3 hours

Topics:

History and Terminology

Security Mindset (reasoned paranoia)

Design Principles (Defense in Depth)

System/security life-cycle

Security implementation mechanisms

(gates, guards, guns; cryptography)

Information assurance analysis model

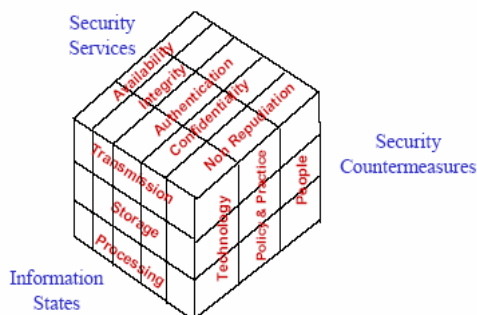
(MSR model*; threats; vulnerabilities; attacks; countermeasures)

Disaster recovery (natural and man-made)

Forensics

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

Information Assurance Model



Core learning outcomes:

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

Elective learning outcomes:

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

IAS2. Security Mechanisms (Countermeasures)

Minimum core coverage time: 5 hours

Topics:

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

Core learning outcomes

1. 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 from the perspectives of accuracy, intrusiveness and efficiency.
6. Describe the differences between symmetric and asymmetric cryptosystems, e.g., number of keys required, the types of algorithms used, etc.
7. Explain what is meant by integrity, confidentiality, and authentication.
8. Describe how cryptosystems offer 1) Confidentiality, and 2) Authentication
9. Describe digital signatures and certificates
10. Describe how a public key infrastructure (PKI) works
11. Describe the DES and 3DES algorithms
12. Demonstrate how public-key cryptography works by the use of public and private keys
13. Describe the AES algorithm
14. Describe the differences between block and stream cryptosystems.
15. Explain the differences in efficiency and performance between software based and hardware based cryptosystems

Elective learning outcomes:

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

IAS3. Operational Issues

Minimum core coverage time: 3 hours

Topics:

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

Core learning outcomes:

1. Describe legal and ethical considerations related to the handling and management of enterprise information assets
2. Specify what constitutes admissible evidence in a legal proceeding and how to acquire and maintain this information.
3. Describe the importance of and key elements involved in incident tracking to develop an incident handling and reporting process
4. Identify risks associated with disasters or disruptions and specify key mitigation strategies.
5. Identify the types of company assets to be protected by a security plan.
6. Specify the key aspects of physical site security.

7. Describe the elements contributing to the cost of an organizations security management and operations process and their relation to risks and losses associated with information assurance or security related issues and incidents.
8. Describe and evaluate employment policies and practices that are relevant to safeguarding an organization's information assets.
9. Describe the importance of utilizing standards and key standard processes currently utilized in information assurance and their areas of relevance (i.e. DES – Data Encryption Standard).
10. Describe the purpose and elements of the key types of security audits. Discuss how various security standards (i.e. ISO 177799) impact the direction of these audits.

Elective learning outcomes:

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

Topics:

- Creation of Policies
- Maintenance of Policies
- Prevention
- Avoidance
- Incident Response (Forensics)
- Domain integration (physical, network, internet, etc.)

Core learning outcomes:

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

8. Describe a situation in which an incident would require a full forensic approach including evidence gathering, full chain of custody auditing and expert analysis.
9. Describe how failure to follow good forensic procedures could make prosecution of an attacker impossible.

Elective learning outcomes:

1. Create a set of policies that implement a specified organizational objective.
2. Update a set of policies to reflect a change in organizational objectives.

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. Explain and give examples of the 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.
4. Describe how the different protocol attacks (e.g. TCP/IP) works against an organization's network.
5. Explain some techniques used during an active attack.
6. Explain some 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 and explain how a Buffer Overflow Attack might be used to compromise a system.
9. Identify and distinguish between the different types of Malware (Viruses, Trojan Horses, Worms).

Elective learning outcomes:

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

IAS6. Security Domains

Minimum core coverage time: 2 hours

Topics:

Human-Computer Interaction
Information Management
Integrative Programming
Networking
Program Fundamentals
Platform Technologies

System Administration
System Integration and Architecture
Social and Professional Issues
Web Systems
Physical plant

Core learning outcomes:

1. Give examples of shared concerns across a specified set of security domains.
2. Give examples of concerns that are specific to specified security domains.

Elective learning outcomes:

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

IAS7. Forensics

Minimum core coverage time: 1 hour

Topics:

Legal Systems
Digital Forensics and its relationship to other Forensic disciplines
Rules of Evidence
Search and Seizure
Digital Evidence
Media Analysis

Core learning outcomes:

1. List three types of legal systems used by countries in the world.
2. Describe how digital forensics fits with the other forensic disciplines.
3. Describe a method for capturing a disk drive as evidence and proving its integrity for legal evidence purposes.
4. Explain the difference between the rules for a corporation seizing its property from an employee and law enforcement's seizing of property from a citizen.

Elective learning outcomes:

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

IAS8. Information States

Minimum core coverage time: 1 hour

Topics:

Transmission
Storage
Processing

Core learning outcomes:

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

Elective learning outcomes:

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

IAS9. Security Services

Minimum core coverage time: 1 hour

Topics:

Availability
Integrity
Confidentiality
Authentication (source reliability)
Non-repudiation

Core learning outcomes:

1. Describe the possible availability levels for a web service.
2. Describe how redundancy and geographic dispersion relate to availability.
3. Define integrity as a security service.
4. Describe how one-way cryptographic functions are used to implement integrity in document transfer.

5. Define confidentiality as a security service.
6. Describe how cryptographic encryption algorithms are used to implement confidentiality in document transfer.
7. Define authentication as a security service.
8. Describe how one-way functions and encryption are used to implement a typical authentication service.
9. Define non-repudiation as a security service.
10. Describe how one-way functions are used to implement a non-repudiation service.

Elective learning outcomes:

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

IAS10. Threat Analysis Model

Minimum core coverage time: 1 hour

Topics:

Risk assessment
Cost benefit

Core learning outcomes:

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

Elective learning outcomes:

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

IAS11. Vulnerabilities

Minimum core coverage time: 1 hour

Topics:

Perpetrators
Inside attacks
External attacks
Black Hat
White Hat

Ignorance
Carelessness
Network
Hardware (design, implementation, installation, etc.)
Software (design, implementation, installation, etc.)
Physical access

Core learning outcomes:

1. Explain the differences between a hacker and a cracker. Consider white hat, black hat, culture, community, tools, and technologies used and how they are used by each.
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. Give an example of how inside and external attacks are similar and are different.
5. List and explain the major network threats and vulnerabilities for an organization.
6. Demonstrate how software contributes to the vulnerabilities for an organization. Consider design, implementation, and installation issues.
7. Demonstrate how hardware contributes to the vulnerabilities for an organization. Consider design, implementation, and installation issues.

Elective learning outcomes:

1. Demonstrate how you would test a system for vulnerabilities.
2. Do a vulnerability analysis of a system.
3. Develop user education modules to educate users on their role in information assurance.
4. Choose a common vulnerability and describe how an attacker uses this vulnerability to gain access to a system.

Information Management (IM) – 34 core hours

IM1. Information Management Concepts and Fundamentals

IM2. Database Query Languages

IM3. Data Organization Architecture

IM4. Data Modeling

IM5. Managing the Database Environment

IM6. Special Purpose Databases

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

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

Elective 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: 9 Hours

Topics:

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

Core learning outcomes:

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

Elective learning outcomes:

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

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
- Star schema
- Normal forms
 - Functional dependencies
 - 1NF
 - 2NF
 - 3NF
 - BCNF
 - 4NF - multi-valued dependencies
 - 5NF - join dependencies
 - Domain Key NF
 - Second order relations
- Referential integrity
- Entity integrity
- Integrity Constraints

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. Label 1NF, 2NF, and 3NF violations given a set of relations and a set of functional dependencies;
7. Demonstrate the decompositions necessary to remove normal form violations for specified 2NF and 3NF violations;
8. Explain how having normal form relations reduces or eliminates attribute redundancy and update/delete anomalies;
9. Define entity integrity;
10. Define referential integrity;
11. Give examples of user defined integrity constraints.

Elective learning outcomes:

1. Given a universal relation R and a set of functional dependencies F , produce a set of relations R such that R contains no BCNF violations or explain why this is not possible for the example;
2. Describe the features added to object-relational systems to distinguish them from standard relational systems.

IM4. Data Modeling

Minimum core coverage time: 6 hours

Topics:

- Conceptual Models
 - Entity Relationship diagrams
 - Enhanced Entity Relationship diagrams
 - Identification of business rules
- Logical Models
- Physical Models
- Reengineering of Databases
- Standardized Modeling in IDEF1, UML
 - Patterns and standard models
- CASE Tools
- Meta-modeling
- Data integration
 - Data warehouses
 - Data marts

Core learning outcomes:

1. Describe and interpret Entity Relationship diagrams;
2. Design a simple Entity Relationship diagram;
3. Describe and interpret Enhanced Entity Relationship diagrams;
4. Identify business rules;
5. Describe a logical model;
6. Describe a physical model;
7. Identify patterns and standard models;
8. Demonstrate an understanding of CASE tools, their usage and application;
9. Describe data integration;
10. Describe meta-modeling;
11. Describe a data warehouse, its basic structure, etc.;

Elective learning outcomes:

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

IM5: Managing the Database Environment

Minimum core coverage time: 3 hours

Topics

- Data Administration
- Database Administration
 - Concurrency
 - Security
 - Backup and recovery
- Distributed databases
 - Types (Homogeneous, heterogeneous, federated)
- Patterns of distribution (Replication, Horizontal/vertical Partitioning)
- Client-server databases
- n-tier architectures
 - Database connectivity
 - ODBC
 - JDBC
 - XML
 - Web services
 - SOAP

Core learning outcomes:

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

Elective learning outcomes:

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

IM6: Special Purpose Databases

Minimum core coverage time: 1 Hour

Topics:

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

Core learning outcomes:

1. Demonstrate an understanding of on-line analytical processing and data warehouse systems.

Elective learning outcomes:

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

Integrative Programming and Technologies (IPT) – 23 core hours

IPT1. Intersystems Communications

IPT2. Data Mapping and Exchange

IPT3. Integrative Coding

IPT4. Scripting Techniques

IPT5. Software Security Practices

IPT6. Miscellaneous Issues

IPT7. Overview of Programming Languages

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

IPT1. Intersystem Communications

Minimum core coverage time: 5 hours

Topics:

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

Core learning outcomes:

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

Elective learning outcomes:

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

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

IPT2. Data Mapping and Exchange

Minimum core coverage time: 4 hours

Topics:

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

Core learning outcomes:

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

Elective learning outcomes:

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

IPT3. Integrative Coding

Minimum core coverage time: 4 hours

Topics:

Design Patterns
Interfaces
Inheritance

Core learning outcomes:

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

Elective learning outcomes:

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

IPT4. Scripting Techniques

Minimum core coverage time: 4 hours

Topics:

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

Core learning outcomes:

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

Elective learning outcomes:

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

IPT5. Software Security Practices

Minimum core coverage time: 4 hours

Topics:

Evidence-Based Security vs. Code Access Security
 Best security coding practices
 Authentication to system resources and services
 Encryption of data between systems and services

Core learning outcomes:

1. Contrast Evidence-Based Security vs. Code Access Security
2. Define the goals of secure coding
3. Give guidelines for authenticating and defining permissions to systems services and resources
4. For each of the following “best secure coding” practices, give an example of a problem that can occur when the practice is not followed and then describe how to overcome the problem:
 - a. Preventing buffer overflow
 - b. Securing State Data
 - c. Securing Method Access
 - d. Wrapper Code
 - e. Unmanaged Code

- f. Validation of user Input
- g. Remoting Considerations
- h. Protected Objects
- i. Serialization
- j. Robust error handling

Elective learning outcomes:

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

IPT6. Miscellaneous Issues

Minimum core coverage time: 1 hours

Topics:

Adopt & Adapt vs. make
Versioning and version control

Core learning outcomes:

1. List the issues that should be considered when creating new software vs. adapting existing software to solve a problem.
2. Tell why it is important to version software and describe one mechanism that can be use to control the versioning of software.

Elective learning outcomes:

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

IPT7. Overview of Programming Languages

Minimum core coverage time: 1 hour

Topics:

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

Core learning outcomes:

1. Contrast the differences between the structured and object-oriented programming paradigms
2. Diagram and label models for both a compiled program and an interpretative program.
3. Describe the benefits and weakness associated with using a virtual machine.
4. Give an example where an application language and a scripting language would be most appropriate, then give a valid rational to support each choice.

Networking (NET) – 20 core hours

NET1. Foundations of Networking

NET2. Routing and Switching

NET3. Physical Layer

NET4. Security

NET5. Application Areas

NET6. Network Management

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. Recognize the names of networking standards bodies and tell what each does.
2. Describe why standards bodies are essential in networking.
3. Give the OSI networking model and describe why it is important.
4. Give a model of the Internet and describe the functions of the components of the model.
5. Identify various communication and terminal equipment used in networking (nodes).
6. Identify the communication channels and media used by various communication equipment in networking (links).
7. Explain how LAN technology is used in the organizational context.
8. Explain how WAN technology is used in the organizational context.
9. Identify and explain the similarities and differences between LANs and WANs.
10. Distinguish between LAN and WAN technologies.
11. Identify and explain the concept of throughput as it pertains to a data communications channel.
12. Identify and explain the concept of bandwidth as it pertains to data communication technology.
13. Distinguish the differences between throughput and bandwidth.
14. Identify the relationship between throughput and bandwidth.
15. Identify different network topologies.
16. Explain the importance of communication protocols.
17. Analyze and compare the use of various communication protocols in different applications.

Elective learning outcomes:

None presently defined

NET2. Routing and Switching

Minimum core coverage time: 8 hours

Topics:

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

Core learning outcomes:

1. Summarize and describe the data communications and telecommunications models, topologies, protocols, standards and architectures in use today.
2. Discuss the concepts and the “building blocks” of today’s data communication networks such as switches, routers, and cabling.
3. Explain the operation 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 terms latency, response time, and jitter and describe their impact on network performance.

Elective learning outcomes:

1. Construct multiple networks and connect them together.
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. Analyze 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, trunking and port aggregation and configure and deploy switches utilizing the above.
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: 6 hours

Topics:

- Wireless & mobile links
- Switched vs packets
- Physical media
- Shannon’s law
- Satellite communication

Error detection and correction
Communication standards
Compression & decompression
Electronics
IEEE 802
Topologies

Core learning outcomes:

1. Describe the three variables of Shannon's law and how they impact channel capacity.
2. List several types of physical communication media, and compare their bandwidth characteristics.
3. Compare and contrast the historical evolution of the switched and routed infrastructures.
4. Summarize the physical challenges inherent to wireless-fixed and wireless-mobile communication channels.
5. Compare and contrast the following methods of error handling: parity, CRC, EDC.
6. Describe how most modern communication standards are developed, addressing both *de jure* and *de facto* standards.
7. Describe how digital data can be compressed.
8. Contrast lossy compression and lossless compression.
9. Describe the function of the IEEE 802 standards committee, and list some of their more widely-known standards.
10. Compare four networking topologies in terms of robustness, expandability, and throughput.

Elective learning outcomes:

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

NET4. Security

Minimum core coverage time: 2 hours

Topics:

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

Core learning outcomes:

1. Explain how secret-key algorithms protect data confidentiality during transport over a network.
2. Explain how public-key algorithms are used in the network for authentication.
3. Explain how public-key algorithms are used in the network to negotiate secret keys.

4. Explain how SSL protects confidentiality of a TCP connection.
5. Describe how a remote access VPN works.
6. Describe how a point-to-point VPN works.
7. Describe how firewalls mitigate some network attack scenarios.
8. Explain why weak passwords are a significant problem in networks.
9. Describe a scenario where an IDS could detect a password cracking attempt.
10. List some of the additional security concerns inherent to wired, wireless and mobile networking.

Elective learning outcomes:

1. Install and configure a firewall.
2. Demonstrate that a firewall is properly configured using a vulnerability testing tool.
3. Install and configure an IDS.
4. Demonstrate the function of an IDS by analyzing a simulated intrusion.

NET5. Application Areas

Minimum core coverage time: 1 hour

Topics:

Multimedia data technologies
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.

Elective learning outcomes:

1. List some of the existing multimedia data technologies.
2. Compare and contrast some of the characteristics of present multimedia data technologies.

NET6. 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.

Elective learning outcomes:

1. Compare and contrast various network management techniques as they apply to wired and wireless networks. (This will include topics on devices, users, quality of service, deployment, and configuration of these technologies.)
2. Dissect different security tools and describe implementation strategies of these tools to ensure efficient and effective protection for wired and wireless networks.

3. Describe the configuration and deployment strategies of network devices to optimize network security.
4. Design and implement a local area network.
5. Design a wide area network.
6. Configure the network nodes (computers, routers, etc).
7. Troubleshoot a network problem.
8. Develop and implement a backup and disaster recovery plan as it relates to the network components of an organization.
9. Develop and implement a security plan as it relates to the network components of an organization.
10. Select, install and use appropriate network management tools to collect performance data from a set of network nodes.

Programming Fundamentals (PF) – 38 core hours

PF1. Fundamental Data Structures

PF2. Fundamental Programming Constructs

PF3. Object-Oriented Programming

PF4. Algorithms and Problem-Solving

PF5. Event-Driven Programming

PF6. Recursion

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: 10 hours

Topics:

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

Core learning outcomes:

1. 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, linked lists, stacks, and queues.
4. Choose the appropriate data structure for modeling a given problem.
5. Describe a simple hash function (priority C level 1)

PF2. Fundamental Programming Constructs

Minimum core coverage time: 9 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

Core learning outcomes:

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

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 subclasses
- Inheritance (overriding, dynamic dispatch)
- Polymorphism (subtype polymorphism vs. inheritance)
- Class hierarchies
- Collection classes and iteration protocols

Core learning outcomes:

1. Discuss and identify the concepts of encapsulation, abstraction, inheritance, and polymorphism.
2. Design, implement, test, and debug simple programs in an object-oriented programming language.
3. Describe how the class mechanism supports encapsulation and information hiding.
4. Design, implement, and test the implementation of “is-a” relationships among objects using a class hierarchy and inheritance.
5. Compare and contrast the notions of overloading and overriding methods in an object-oriented language.
6. 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 algorithms for solving simple problems.
4. Use pseudo-code or a programming language to implement, test, and debug algorithms for solving simple problems.
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 simple event-driven programs that respond to user events.
3. Develop code that responds to exception conditions raised during execution.

PF6. Recursion

Minimum core coverage time: 1 hours

Topics:

- The concept of recursion

Core learning outcomes:

1. Describe the concept of recursion and give examples of its use.

Platform Technologies (PT) – 14 core hours

PT1. Operating Systems

PT2. Architecture and Organization

PT3. Computing infrastructures

PT4. Enterprise Deployment Software

PT5. Firmware

PT6. Hardware

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: 10 hours

Topics:

- Overview
- Windows versus Unix-class Operating Systems
- 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. Use both Windows and Unix-class systems.
3. Describe the similarities and differences between Windows and Unix-class systems.
4. Explain the benefits of using scripts to perform operating systems tasks.

Elective learning outcomes:

1. Analyze operating system requirements and recommend an appropriate operating system to meet the requirements.
2. Install a Windows operating system.
3. Install a Unix-class operating system.
4. Plan 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
Functional organization
Multiprocessing and alternative architectures
Performance enhancements

Core learning outcomes:

1. Describe how numbers and characters are represented in a computer.
2. Draw a block diagram, including interconnections, of the main parts of a computer.
3. Describe how a computer stores and retrieves information to/from memory and hard drives.
4. Define the terms: bus, handshaking, serial, parallel, data rate.
5. Note to self: what does *functional organization* mean here?

Elective learning outcomes:

1. Write an assembly-language program which will take two numbers from the keyboard, add them in binary, and send the result to the screen.
2. Write an assembly-language program which will do a sort on ten memory locations.
3. Write a program which will perform hard-disk defragmentation.

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. Explain 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.

Elective learning outcomes:

1. Prepare a computer system for use as a server.
2. Install a multi-computer network and deliver computing services on that network.
3. Explain the benefits of using server farms.

PT4. Enterprise Deployment Software

Minimum core coverage time: 0 hours

Topics:

Middleware frameworks
Enterprise deployment and management software
Configuration, definition and management

Core learning outcomes:

None. The coverage time for Everyone is 0 hours.

Elective learning outcomes:

None presently defined.

PT5. Firmware

Minimum core coverage time: 0 hours

Topics:

Firmware storage options
Firmware, software, and hardware
Versioning

Core learning outcomes:

None. The coverage time for Everyone is 0 hours.

Elective learning outcomes:

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

PT6. Hardware

Minimum core coverage time: 0 hours

Topics:

Digital logic & digital systems
Benchmarking
Serial vs parallel
Implementation options
Basic electronics
Hardware design languages
Virtual machine emulation

Core learning outcomes:

None. The coverage time for Everyone is 0 hours.

Elective learning outcomes:

1. Design a simple finite state machine with at least 6 states and 4 conditional branches, then build and troubleshoot it.
2. Complete a comparison in the performance of two different computers with two different operating systems.
3. List the advantages and disadvantages of the five main hardware implementation options.

System Administration and Maintenance (SA) – 11 core hours

SA1. Operating Systems

SA2. Applications

SA3. Administrative Activities

SA4. Administrative Domains

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

SA1. Operating Systems

Minimum core coverage time: 4 hours

Topics:

- Installation
- Configuration
- Maintenance (service packs, patches, etc.)
- Server services (print, file, DHCP, DNS, FTP, HTTP, mail, SNMP, telnet)
- Client services
- Support

Core learning outcomes:

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

Elective 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. Analyze the pros and cons of installing service packs and patches.
4. Recommend when service packs and operating system patches should be installed.
5. Install service packs and operating system patches.
6. Install various server and client services.
7. Demonstrate the ability to support various operating systems.

SA2. Applications

Minimum core coverage time: 3 hours

Topics:

- Installation
- Configuration

Maintenance (service packs, patches, etc.)
Server services (database, web, network services, etc.)
Client services
Support

Core learning outcomes:

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

Elective learning outcomes:

None presently defined.

SA3. Administrative Activities

Minimum core coverage time: 2 hours

Topics:

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

Core learning outcomes:

1. Describe the benefits of content management within an organization.
2. Discuss the need for content deployment.
3. Identify and explain the responsibilities associated with server administration and management.
4. Describe the benefits of managing users and groups.
5. Identify the need for managing IT resources.
6. Describe the benefits of automation management.
7. Identify situations in which administrative activities are required.
8. Identify situations which interfere with administrative activities.
9. Identify the need for policies governing of IT systems.
10. Identify the need to train users on IT systems and policies.
11. Identify the need for policies governing of IT systems.
12. Identify the need to train users on IT systems and policies.

Elective learning outcomes:

1. Identify the need for policies governing of IT systems.
2. Identify the need to train users on IT systems and policies.
3. Develop policies governing IT systems.
4. Design and deliver training sessions on IT systems and policies.

SA4. Administrative Domains

Minimum core coverage time: 2 hours

Topics:

Web domain
Network domain
Database domain
OS domain
Support domain

Core learning outcomes:

1. Identify the responsibilities common to the various administrative domains.
2. Identify the differences in the responsibilities of the various administrative domains.

Elective learning outcomes:

None presently defined.

System Integration and Architecture (SIA) – 21 core hours

SIA1. Requirements

SIA2. Acquisition / Sourcing

SIA3. Integration

SIA4. Project Management

SIA5. Testing and QA

SIA6. Organizational Context

SIA7. Architecture

One of the roles of the IT professional is to design and build systems and integrate them into an organization. This knowledge area develops the skills to gather requirements, 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: 6 hours

Topics:

- Gathering user, task, technical
- Modeling
- Testing
- Lifecycle

Core learning outcomes:

1. Survey the user environment for appropriate tasks.
2. Classify the different users and their characteristics.
3. Explain the various requirements modeling techniques.
4. Explain how requirements gathering fits into a system development lifecycle.
5. Explain the different types of testing needed for the requirements phase of the lifecycle.
6. Describe the problems that can arise if correct testing requirements are not established for system acceptance.
7. Outline the requirements section of an RFP.

Elective learning outcomes:

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

SIA2: Acquisition / Sourcing

Minimum core coverage time: 4 hours

Topics:

- Build / buy
- In-sourcing/outsourcing
- Hardware

Software
Support
Testing / evaluation / benchmarking
Contracts/RFP
Quality

Core learning outcomes:

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

Elective learning outcomes:

1. Analyze the appropriateness of a decision to buy or build a software package or a hardware component in a given situation;
2. Analyze the appropriateness of a decision to in-source or out-source IT services in a given situation;
3. Given a set of evaluation criteria, test, evaluate and/or benchmark an IT system, or a component thereof;
4. Examine a given RFP and comment on how well-structured it is;
5. Examine a given contract and comment on how well-structured it is.

SIA3: Integration

Minimum core coverage time: 3 hours

Topics:

Components, interfaces and integration
Infrastructure, Middleware and Platforms
Techniques – Data warehouses, extending frameworks, wrappers, glue, facades
Testing/evaluation/benchmarking

Core learning outcomes:

1. Define integration in terms of components and interfaces.
2. Give examples of middleware platforms.
3. List some advantages and disadvantages of some middleware platforms.
4. List some major considerations for enterprise integration platform selection.
5. Give an example of integration using the “wrapper” approach.
6. Give an example of integration using the “glue code” approach.
7. Give an example of how a framework facilitates integration of components.
8. Explain how the data warehouse concept relates to enterprise information integration.
9. Give examples of how testing and evaluation are impacted by integration choices.

Elective learning outcomes:

1. Install and test an enterprise integration middleware platform.
2. Develop a component and demonstrate its integration into an existing environment.

SIA4. Project Management

Minimum core coverage time: 3 hours

Topics:

- Cost Benefit analysis
- 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.
2. Describe the importance of a cost/benefit analysis to the successful implementation of a project plan.
3. Describe roles and responsibilities for key project personnel and stakeholders.
4. Use appropriate project planning and tracking tools.
5. Describe issues involved in creating a project schedule.
6. Describe the purpose of a project post-mortem (closeout) and review session.

Elective learning outcomes:

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

SIA5. Testing and QA

Minimum core coverage time: 3 hours

Topics:

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

Core learning outcomes:

1. Define the current testing standards.
2. Explain the various components of usability testing.
3. Describe the techniques used in testing a system or product.
4. Identify appropriate acceptance criteria.

Elective learning outcomes:

1. Design a stress test that will test and pinpoint the critical issues that might impact system performance.
2. Compare the results of a usability test to determine if the product meets the stated acceptance criteria.
3. Summarize the data from a usability test and make appropriate recommendations.
4. Select the appropriate techniques to be used in a testing environment.

SIA6: Organizational Context

Minimum core coverage time: 1 hour

Topics:

Business processes
IT environment
Organizational Culture

Core learning outcomes:

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

Elective learning outcomes:

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

SIA7: Architecture

Minimum core coverage time: 1 hour

Topics:

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

Core learning outcomes:

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

Elective learning outcomes:

1. Describe the major features of a Customer Resource Management system.
2. Describe the major features of an Enterprise Resource Planning system.

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

Social and Professional Issues (SP) – 23 core hours

- SP1. Professional Communications**
- SP2. History of Computing**
- SP3. Social Context of Computing**
- SP4. Teamwork Concepts and Issues**
- SP5. Intellectual Property**
- SP6. Legal Issues in Computing**
- SP7. Organizational Context**
- SP8. Professional and Ethical Issues & Responsibilities**
- SP9. Privacy and Civil Liberties**

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

SP1. Professional Communications

Minimum core coverage time: 5 hours

Topics:

- Technical memos
- System documentation
- Technical requirements

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 technical memo to management.
4. Create user documentation for an IT system.
5. Create a set of technical requirements for an IT system.
6. Compare and contrast technical writing with expository writing.

SP2. History of Computing

Minimum core coverage time: 3 hours

Topics:

- Implications of:
 - History of computer hardware, software
 - History of the Internet
 - Telecommunications
- The IT profession
- IT education

Core learning outcomes:

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

SP3. Social Context of Computing

Minimum core coverage time: 3 hours

Topics:

- Social informatics
- Social impact of IT on society
- Online communities & social implications
- Philosophical context
- Diversity issues
- Accessibility
- 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 and diversity related 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. Teamwork Concepts and Issues

Minimum core coverage time: 3 hours

Topics:

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

Core learning outcomes:

1. Compare and contrast personality types and their affect on creating better teams.
2. Describe the basic elements of group dynamics.
3. Compare and contrast different conflict resolution strategies.
4. Contrast and compare basic leadership styles and how they affect teams.
5. Summarize and apply the elements of groupware and how it can be used for team projects.
6. Describe ways in which collaboration can be used to assist in cross-functional teams.
7. Apply cross-functional team reports when developing IT products.

SP5. Intellectual Properties

Minimum core coverage time: 2 hours

Topics:

- International differences
- Digital Millennium Copyright Act (DMCA)
- TEACH Act**

Ownership of information
Plagiarism
Copyright, patent, and trademark laws
Trade secrets

Core learning outcomes:

1. Distinguish among copyright, patent, and trademark laws.
2. Discuss the ramifications of trade secret contracts.
3. Discuss the implications of plagiarism both in education and the profession.
4. Discuss how patent and copyright laws may vary internationally.
5. Describe the areas of the Digital Millennium Copyright Act.
6. Describe the areas of the TEACH Act.

SP6. Legal Issues in Computing

Minimum core coverage time: 2 hours

Topics:

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

Core learning outcomes:

1. Outline the technical basis of viruses and denial-of-service attacks.
2. Explain the pros and cons of computer hacking and cracking.
3. Discuss the types of policies that should be included for system use and monitoring.
4. Describe the basis elements of compliance laws – such as ADA508, FERPA and HIPPA.
5. Demonstrate how an application satisfies a particular compliance law.
6. Describe the differences in accountability, responsibility, and liability.
7. Discuss the consequences of software piracy on software developers and the role of relevant enforcement organizations.
8. Discuss how risk can be managed in an information technology environment.
9. Produce a computer use policy with enforcement measures.

SP7. Organizational Context

Minimum core coverage time: 2 hours

Topics:

Business processes
IT environment
Organizational culture
Professionalism

Core learning outcomes:

1. Outline the basic parts of a typical IT environment.
2. Explain how IT must support business processes.

3. Identify how an IT professional maintains their professional behavior.
4. Explain how an organizational culture can affect IT.

SP8. Professional and Ethical Issues & Responsibilities

Minimum core coverage time: 2 hours

Topics:

Relationships with professional societies
Codes of professional conduct, such as IEEE, ACM, BCS, ITAA, AITP
Ethics and history of ethics
Identify theft
Ethical hacking

Core learning outcomes:

1. Identify the strengths and weaknesses of relevant professional codes as expressions of professionalism and guidance to decision-making.
2. Apply appropriate codes in assignments.
3. List the underlying philosophical aspects of ethical decision making.
4. Identify how society has been affected by identify theft and what to do to protect individuals.
5. Discuss the pros and cons of ethical hacking.

SP9. Privacy and Civil Liberties

Minimum core coverage time: 1 hour

Topics:

HIPPA and FERPA
E.U. Data Protection
Gramm-Leach-Bailey Act

Core learning outcomes:

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

Web Systems & Technologies (WS) – 21 core hours

WS1. Web Technologies

WS2. Information Architecture

WS3. Digital Media

WS4. Web Development

WS5. Vulnerabilities

WS6. Social Software

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: 10 hours

Topics:

- HTTP Protocol
- Presentation abstractions
- Web-markup and display languages
- Client-side programming
- Server-side programming
- Web services
- Web servers
- Emerging technologies
- Standards & 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. Understand XML syntax and how to display such documents in Web applications.
5. Describe some presentation technologies, such as, Cascading Style Sheets and DHTML.
6. Discuss and contrast data entry and validation techniques in Client-side vs. Server-side programming.
7. Discuss and contrast Client-side with Server-side security issues.
8. Describe the use of server-side backend databases in web sites and web applications.
9. Describe technologies used in Web Services, including Open Source languages and packages, Proprietary languages and packages, and Enterprise Web development and Distributed Web applications.
10. Describe Open Source and Proprietary Server software.
11. Describe the differences between types of web servers including application servers, streaming media servers and transformation servers.
12. List and discuss the function of existing and emerging Web technologies, such as XML, SOAP, WSDL, UDDI, Java, C#, EJB, .NET, etc.
13. Discuss Web standards and standard bodies including the World Wide Web Consortium (W3C).
14. Discuss Web Standards in terms of specifications, guidelines, software, and tools.

Elective learning outcomes

1. Apply and integrate HTML/XHTML/XML syntax to create and validate documents.
2. Apply and integrate HTML/XHTML/XML syntax to generate contents via programming.
3. Apply and integrate XML syntax to transform documents between formats.
4. Apply presentation technologies to author websites, such as Cascading Style Sheets, XSLT-FO, DHTML, etc.
5. Use specialized Web Markup languages, such as SVG, SMIL, RSS, etc.
6. Construct website to include Client-side programming with JavaScript, Java Applets, Flash, and other Web GUI technologies.
7. Discuss data persistence via cookies in maintaining states.
8. Construct Web server programs with Server-side programming using/working with Generic HTTP CGI techniques, Open Source as well as Proprietary languages and packages.
9. Discuss issues and implementation of server-side security.
10. Discuss issues in maintaining backend databases.
11. Design, construct, test and evaluate web services using Open Source languages and packages, proprietary languages and packages and enterprise web development technology.
12. Design, construct, test and evaluate distributed web applications.
13. Contrast and explain the issues involved in designing web servers including web application servers, streaming media servers, and transformation servers.
14. Assess, compare and select emerging and existing Web technologies for solutions, such as XML, SOAP, WSDL, UDDI, Java, C#, EJB, .NET, etc.
15. 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:

1. Hypertext/hypermedia:
 - a. Effective communication
 - b. Interfaces
 - c. Navigation schemes
 - d. Media types
2. Web design process:
 - a. User modeling and user-driven design
 - b. Web Design Patterns
 - c. Information organization
 - d. Usability
 - e. N-Tier architectures

Core learning outcomes:

1. Build a simple web site that organizes information effectively
2. Identify alternative ways to organize and present information on a web site.
3. Select an organization of information based on its inherent structure (chronological, alphabetic, etc.).
4. Identify the purpose of a web site or genre (ecommerce, self-service, educational, governmental service, etc.).
5. Choose a graphic file type that matches the image characteristics and use.
6. Identify time-based media types commonly used on the web.

7. Discuss the user of proprietary media and interaction technologies such as Flash, Active X, RealMedia, and QuickTime.
8. Use cascading style sheets to create style standards for a web site.
9. Describe characteristics of users of a web site that affect design.
10. Describe characteristics that enhance usability of a web site.

Elective 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, branding and simplify development of a site.
6. Design an effective homepage for a site.
7. Conduct usability tests for a web site.
8. Explain separation of concerns as it applies to the design and implementation of a web site.
9. Create client-side web-based user interfaces for navigation.
10. Create server-side web-based user interface for applications.
11. Implement a three-tier web site.

WS3. Digital Media

Minimum core coverage time: 3 hours

Topics:

- Digital Libraries
- Media formats
- Capture, Authoring and Production Tools
- Compression
- Streaming media

Core learning outcomes:

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

Elective learning outcomes:

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

7. Deploy and serve media contents within Web applications.

WS4: Web Development

Minimum core coverage time: 3 hours

Topics:

- Web Interfaces
- Web Site Implementation and Integration
- Database Integration
- Accessibility Issues
 - Web Accessibility Initiative

Core learning outcomes:

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

Elective Learning Outcomes:

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

WS5. Vulnerabilities

Minimum core coverage time: 1 hour

Topics:

- Client Security
 - cookies and web beacons
 - phishing
 - transaction security - certificates and secure connections
 - spyware
 - viruses
 - man-in-the-middle attacks
- Server Security
 - Securing a server
 - Securing host
 - Denial-of-service attacks
 - Attacks via clients (eg zombies and DOS)
 - DNS poisoning

Core learning outcomes:

1. Describe the legitimate uses of cookies.
2. Describe the methods of security for cookies
3. Identify web sites that are using cookies.

4. Identify web sites that are using web page graphics as web beacons.
5. Describe how web beacons circumvent cookie safeguards
6. Name ways in which cookies can be used to compromise user privacy.
7. Describe ways to increase the trustworthiness of a website such as security certificates.
8. Identify web applications that are using an encrypted connection.
9. Describe the use of public key encryption to enhance security.
10. Describe phishing and ways to identify it.
11. Describe spyware and how it compromises user security and privacy.
12. Identify ways that spyware is introduced into a user's computer.
13. Describe viruses and how they are introduced into a user's system.
14. Explain what a denial of service attacks is and how they are done.
15. Explain how to protect clients from viruses, spyware and zombie processes.

Elective learning outcomes:

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

WS6: Social Software

Minimum coverage time: 0

Topics:

Asynchronous and synchronous communication modalities
 Broadcast communication modalities
 Collaborative and community modalities
 Ethical issues
 Digital Divide
 Freedom of Speech vs. Hate Speech, Pornography
 Privacy
 Copy/Digital content rights

Core learning outcomes:

None. The coverage time for Everyone is 0 hours.

Elective 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.



Appendix B Course Descriptions

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

B.1 Curricular Approaches

In Chapter 6 [section 6.3](#), four curricular approaches were described: *practicum first*, *theory first*, *breadth first* and *depth first*. The practicum-first and theory-first strategies are orthogonal to the latter two; an IT curriculum may choose either or a hybrid of the first two strategies, along with either of the second two. An example of the breadth first approach is given in detail below. Examples of the other curricular approaches will follow when they are completed.

Breadth First Approach Introductory Courses

Course #	Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
IT 101	IT Fundamentals	Introduces students to the academic discipline of IT. Pervasive IT themes; IT history; Organizational issues; Relationship of IT to other computing disciplines	None	ITF: 1(17), 2(6), 3(3), 4(3), 5(2), 6(2); SP: 2(1), 3(2)	Pervasive themes in IT; Organizational issues; History of IT; IT and its Related and Informing Disciplines; Application Domains; Applications of Math and Statistics to IT	a; b; f; k
IT/CS 130	Programming Fundamentals	Introduces students to the basics of programming, including data structures, programming constructs, object-oriented programming, algorithms and problem solving, event-driven programming, and recursion.	None	PF: 1(10), 2(9), 3(9), 4(6), 5(3), 6(1)	Fundamental data structures and programming constructs; Object-oriented programming; Algorithms and problem solving; Event-driven programming; Recursion; Overview of programming languages	a; c; g; h
IT/CS 150	Computer Architecture	Principles of computer hardware and low-level software, including logic circuits, assembly language, I/O, storage, program execution	IT 101; IT 130	PT: 2(3), 3(1); SP: 2(1)	Architecture and organization; Computing infrastructures	a; c; e; g; h
IT/CS 160	Operating Systems	Basics of computer operating systems, including configuration, file systems, security, administration, interfacing, multitasking, performance analysis	IT 101; IT 130	PT: 1(10); SA: 1(3); SP: 2(1), 3(1)	Operating systems	a; b; c; e; g; h

Intermediate Courses

Course #	Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
IT 210	IT Systems	Introduction to the basic components of IT systems, including networking, web systems, databases, scripting, system administration and maintenance, and system integration	IT 101; IT 130	HCI: 1(2), 2(1), 3(1), 4(1), 5(1); IAS: 1(1), 2(1), 5(1), 6(1), 7(1), 8(1), 9(1), 10(1); IM: 1(1), 2(4), 3(1), 4(2); IPG: 1(3), 2(3), 3(1), 4(1), 6(1); NET: 1(1), 2(1), 3(1), 5(1); PF: 5(1); PT: 1(2), 2(1); SA: 1(1); SIA:3(2); WS: 1(6), 2(2), 3(3), 4(3). (Total Hours=56)	Foundations of networking; Database systems; Data administration; Database administration; Web technologies; Scripting techniques; Integrative coding; Applications; Integration; History of computing	a; b; c; e; f; g; h; k
IT 250	Web Systems	Introduction to web technologies and systems, including hypertext, self-descriptive text, web page design, web navigational systems, and digital media	IT 210	WS: 1(4), 2(2), 3(1), 4(1), 5(1)	Web technologies; Information architecture; Digital media; Web development; Vulnerabilities	a; b; c; e; f; g; h; i; j
IT 330	Networking	Builds a deeper understanding of how networks work, including the topics of LANs, WANs, service providers, packets, hubs, routers, switches, Internet protocols	IT 210	NET: 1(7), 2(5), 3(2), 4(2)	Routing and switching; Physical layer; Security; Application Areas	a; b; c; e; f; g; h
IT 350	Databases	Builds a deeper understanding of how databases work, including the topics of database theory and architecture, data modeling, normalization, query languages, security, and Web applications	IT 210	IM: 1(8), 2(6), 3(6), 4(1), 5(1), 6(1)	Data organization and retrieval; Data modeling; Database query languages; Database systems; Information models and systems; Properties of data; Specification of data requirements; Data and database administration	a; b; c; e; f; g; h; i; j
IT 370	Human-Computer Interaction	Introduction to the basic concepts of human-computer interaction, including human factors, performance analysis, cognitive processing, usability studies, environment, and training	IT 210	HCI: 1(4), 2(2), 3(2), 4(2), 5(1), 6(1), 7(2)	Human factors; HCI aspects of application domains; Human-centered evaluation; Developing effective interfaces; Emerging technologies; Human-centered software; Accessibility	a; b; c; e; f; g; h; i; j
ENGL 310	Professional Communication	Introduction to written and oral technical and professional communication, including proposals, reports, presentations, formal papers	College English	SP: 1(25)	Professional Communications	j; k

Advanced Courses

Course #	Title	Description	Prerequisites	KAs Covered	Units Covered	Program Outcomes
IT 410	Information Assurance & Security	Introduction to the concepts of data security, including policies, attacks, vulnerabilities, encryption, information states, and forensics	IT 330, IT 350, IT 250	IAS: 1(4), 2(2), 3(3), 4(3), 5(1), 6(1), 7(1), 8(1), 9(1), 10(1), 11(1)	Security mechanisms; Fundamental aspects; Security services; Information states; Threat analysis model; Vulnerabilities; Attacks; Policy; Operational issues; Forensics; Security domains	a; b; c; e; f; g; h;
IT 480	IT Capstone I	IT senior project-first semester, including project proposal, feasibility studies, intellectual property, teamwork, budgets, schedule management	IT 330, IT 350, IT 370	SIA: 1(6), 2(4), 3(1), 4(2), 6(1), 7(1); SP: 1(2), 4(3), 5(2), 6(2), 7(2), 8(2), 9(1)	Requirements; Acquisition/sourcing; Integration; Project management; Architecture; Teamwork concepts and issues; Intellectual properties; Organizational context	a; b; c; d; e; f; g; h; i; j; k
IT 481	IT Capstone II	IT senior project-second semester, including teamwork, professional communications (reports and presentations), design implementation, testing	IT 480	SA: 2(3), 3(2), 4(2); SIA: 3(2), 4(1), 5(3); SP: 1(3), 4(1)	Testing and QA; Organizational context; Professional communications; Teamwork concepts and issues;	a; b; c; d; e; f; g; h; i; j; k
IT 490	Professional Ethics	Covers all the areas of ethics in the computing profession	IT 330, IT 350, IT 370	SP: 8(2)	Professional and Ethical Issues and Responsibilities; Privacy and Civil Liberties	b; f

Blue=All KA Covered

Red=All Unit Covered



Appendix C

Learning Outcomes/Knowledge Areas Matrix

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

LEARNING OUTCOME	KNOWLEDGE AREAS/UNITS
(a) Use and apply current technical concepts and practices in the core information technologies;	ITF; HCI 1-4,7; IAS; IM; IPT; NET; PF; PT; SIA7; SP2; WS1; WS3-5; PerTheme1,2,4
(b) Analyze, identify and define the requirements that must be satisfied to address problems or opportunities faced by organizations or individuals;	ITF6; HCI7, IAS; IM4; SIA1,2: WS2; PerTheme1,2
(c) Design effective and usable IT-based solutions and integrate them into the user environment	HCI4; IM4; IPT6; NET6; PT3 (PT5,6); SA; SIA2,3,5,6; WS2; PerTheme2
(d) Assist in the creation of an effective project plan;	ITF2,6; SIA4; PerTheme3
(e) Identify and evaluate current and emerging technologies and assess their applicability to address the users' needs;	HCI6; WS1; PerTheme2
(f) Analyze the impact of technology on individuals, organizations and society, including ethical, legal and policy issues;	HCI 1, 4, 5, 7; IAS3,7,10,11; IM1; NET5; SA3; SIA6; SP2,3,5,6,8,9: (WS6); PerTheme2,6
(g) Demonstrate an understanding of best practices and standards and their application;	ITF1; HCI3; IAS 2, 3, 10; IPT2,5; NET1-3; SIA7; SP8; WS1,3; PerTheme2,3
(h) Demonstrate independent critical thinking and problem solving skills;	HCI1, PF
(i) Collaborate in teams to accomplish a common goal by integrating personal initiative and group cooperation;	(SIA4); SP4; PerTheme7
(j) Communicate effectively and efficiently with clients, users and peers both verbally and in writing, using appropriate terminology;	ITF1; SP1,6; PerTheme7
(k) Recognize the need for continued learning throughout their career.	ITF; PerTheme5,6