

IT Curriculum Proposal – Four-year Degrees

This document outlines the current state of the IT curriculum developed by the SITE Curriculum Committee. This is a working document. The goal of this committee was to develop a curriculum that was non-exclusive, flexible, customizable and could provide for the rapid evolution of the IT discipline without the need to radically overhaul the curriculum periodically. The curriculum was developed in several phases. The rationale for this approach follows.

We first sought to understand where students would begin their education. This defines a common starting point for all IT students making it possible for programs to be readily designed around this least common denominator. Next we considered what students intend to do with their education upon completing a degree. This defined the expected outcomes or ending point of the curricula. Once the beginning and ending points were defined, we designed a curriculum model that would bridge the gap between the common starting point and various endpoints. This model works by providing a template that can be used to design a curriculum for a specific outcome. This provides for flexibility in design and implementation of curricula. The model is based on identifying topic clusters appropriate for the identified outcomes. As educators, we must all feel that the curriculum we are teaching now is a good one. Hence, as a final check, the model will be validated against the current curricula of the member schools in SITE. This provides not only a sanity check, but will provide for an evolutionary implementation path for schools from their present curriculum to the new model rather than an abrupt change.

Phase one involved developing a common core for all IT students that would define the starting point for all IT curricula. Phase two involved identifying possible careers that students would intend to pursue upon graduation. Phase three involved developing topic clusters to support those careers and developing a template for creating curricula for a given outcome. Phase four will involve an evaluation exercise whereby the members of SITE may use the model created here to evaluate their present curricula.

The presentation of the content here will largely follow the phased development. Where credits and hours are mentioned, it is assumed that the term is in the context of semester credits or hours unless specified otherwise.

If the ratification of this model is forthcoming, the remaining work will entail the development of course abstracts, the development of some standard model curricula and the development of procedures for modifying and amending the curriculum document.

Phase 1: The IT Curriculum Core

Currently the document expresses the rationale for each core topic area and the discipline topics that would be covered in each area. The general idea is that each topic area listed (networking, database, programming, web, math and communication) would be a 3-semester-credit course (total of 18 credits) and that these courses would be covered as a foundation to further depth in each area.

This implementation is not cast in stone and it may prove the core can be taught in fewer credits or may require more credits. Ultimately, the emphasis is that the core concepts will be taught and the implementation may be left for individual schools and programs to decide. For the purpose of developing the remainder of the curriculum model, we assume that this core represents 18 credits of

the curriculum and is covered as a prelude to further depth in these areas. The credits assigned here are by no means meant to imply a required implementation.

The content is presented here in terms of topics that need to be presented rather than courses. The topics are presented in the structure of the major topic areas programming, networking, database and web computing.

Problem Solving and Programming

The context is the development of programs to solve problems not the application of off-the-shelf software packages to customer problems.

Objective: Students must be able to solve problems via computer software in a correct, effective and reasonably efficient manner. Students must understand the concepts of software design, program structure, logical design, algorithm efficiency, algorithm development, and algorithm correctness, problem-solving and data representation. Students must understand the concept that the algorithm is the foundational element of program design and independent of high-level design paradigm or programming language. Students must be able to write and debug simple programs using a high-level programming language.

Topics:

- Program design paradigms

 - Structured procedural, object oriented, event driven.

- Primitive operations of a computer

 - Store/retrieve data perform arithmetic; compare two data elements, control execution sequence, input data, and output data.

- Structure Theorem – essential control structures are sequence, repetition and conditional

- Writing algorithms

 - Design, development, correctness, efficiency, pseudocode.

- Data representation

 - Binary coding, data types, variables, constants, and storage.

- Data analysis

 - Verifying and interpreting results.

The intent is that students need to develop the ability to attack a problem and solve it in terms of writing an algorithm. Students must be able to express their algorithm in some generic, easily understood, language-independent manner (pseudocode). The students also need to be familiar with the various design paradigms that can be used to implement the algorithms into a software solution. At a minimum, the student should be aware of the structured procedural, objective oriented and event driven programming paradigms with the ability to incorporate whatever new paradigms may evolve in the future. Students must understand the nature of data, the analysis of data, and the representation of data in the computer. Students must understand that the correctness of a program is not the only measure of a program. In addition to producing correct results, it is important that a program can be maintained (through proper design and internal documentation), can be effectively used by the customer (through interface design, external documentation and help system), and is reasonably efficient. Students must understand the basic capabilities of a computer in terms of its basic (primitive) operations and that with these few basic operations and three logic control structures, any program can be written (Structure Theorem). Students must realize that they will generally be confronted with problems that are not within their previous problems solving

experience. Students must develop the ability to interpret a problem statement, research problem elements new to their experience and apply their research to solving the problem.

We expect that these topics can be taught in one or two three-credit courses depending on the nature of the students and institution.

Networking

The context is the fundamental understanding of networks as communication systems. Emphasis is placed on computer networks and how computer networks and distributed computing are used to solve customer problems.

Objective: Students must be able to identify applications of networks as communication systems. Students must be able to identify computer networks based on the structure, application and size of the network. Students must understand the need, use and operation of network protocols. Students must understand basic system and network security.

Topics:

Applications of networks as communication systems

Business computer networks, CAD/CAM, telecommunications, broadband (cable and satellite), control systems (automotive, manufacturing, environmental, security, monitoring, etc.).

Computer network types

LAN, CAN, MAN, WAN, SAN, enterprise networks (VPN), Internet.

Protocols

OSI model (control concepts and layered operation), TCP/IP, HTTP, FTP.

Distributed systems

Servers, clients, parallel/distributed computing, concurrency, resource deployment/sharing, interdependent applications and systems.

Security

Authorization, encryption, data integrity, backup, disaster recovery, viruses.

Databases

The context is the uses of data processing in solving customer problems.

Objective: Students must understand the concept of data and how information is derived from the data. Students must understand the concept of data integrity (consistency and reliability). Students must understand the various methods for storing and retrieving data. Students must understand how the data will be processed and how the storage method will affect processing. Students must be familiar with the concept of files and databases. Students should understand the role of databases and distributed databases in software applications, particularly, Internet applications. Students must be able to create simple databases and query them using SQL.

Topics:

Data and Information

- Data vs. information, data interpretation, information coding
- Data storage methods
 - Mass storage devices, unstructured storage (files), structured storage (databases), data representation, file access methods (read, read/write, etc.), records, fields, database types (relational, object-relational, etc.), data independence (logical and physical)
- Data integrity
 - Concurrent access, record locking, security, transactions
- Databases
 - Data management, data integrity, synchronization, real-time systems, data mining, queries, SQL
- Distributed databases
 - Distributed network applications, Internet applications, e-commerce.

Web Technologies and Issues

The context is that the student will become adept at using fundamental web technologies to establish a web presence either personal or professional.

Objective: Student should become familiar with fundamental web technologies. Student will understand the Internet's history, structure and operation. The student will understand application of the Internet in business, research and data mining. The student will be able to establish a web presence utilizing effective design and web programming technologies.

Topics:

- The Internet
 - History, structure, operation, protocols, service providers, URL, DNS, domains.
- Applications
 - E-commerce, research, data mining, education, entertainment, enterprise network (VPN).
- Programming
 - Client side, server side, scripting languages, mark-up languages, event driven design.
- Design
 - Color, layout, navigation, site organization, consistency.

General Education

There are two specific areas where components of the general education are specified. If these cannot be accommodated via the general education portion of the school's curriculum, the IT department should undertake designing and delivering the courses in context.

Math

The core math requirement for IT students consists of some basic discrete math and data analysis concepts.

Topics:

Other-base number systems

Binary, octal and hexadecimal.

Set concepts

Sets, elements, membership, subsets, empty set, union, intersection, complement, Cartesian product, power set

Logic

Statements, AND, OR, NOT, XOR, truth tables, DeMorgan's Law.

Number theory concepts

Counting, factorial, combinations, permutations, discrete probability, Fundamental Counting Theorem, prime numbers, composite numbers, GCD, LCM.

Mathematical relations

Ordered pairs, functions, mathematical modeling, input (domain), output (range).

Basic data analysis

Mean, median, mode, mid-range, variance, standard deviation, range.

English

The core English requirement for IT students consists of applying writing and communication skills in a technical environment. These skills should be reinforced throughout the curriculum.

Topics:

Technical writing

Content, readability, grammar, organization, documentation (internal, external, user's guide, etc.).

Technical communication

Verbal presentation skills, professional interpersonal communication.

Phase 2: Expected Curricular Outcomes

The expected outcome of an undergraduate degree program can be classified into one of two broad categories. Generally speaking, a student will either seek further education or seek employment. Since the objective of this committee was to develop an undergraduate curriculum model, the only case of further education explicitly dealt with here is the case of a student completing a two-year degree for transfer to a four-year program. However, the only real requirement for graduate study as opposed to employment is suitable preparation for the selected graduate program. At such time as this curriculum document is extended to cover graduate study in IT, those preparations can be made more concrete.

Following is a list of careers identified by committee members through their own experience and research efforts. The bulk of the careers are listed here within one of the four core IT topic areas indicating the area having the greatest influence on that career. This suggests the major cluster area the student should take to prepare for that particular career. Additionally, there are some careers that are listed here that were not so easily classified into one of the four core areas but rather have a more broad scope.

Networking Job Titles

Cable Installer

Communications Analyst
Customer Service Coordinator
Hardware Installations Coordinator
Hardware Support/Maintenance
Local Area Network Technician
Network Administration
Network Administrator
Network Analyst
Network Maintenance and Operations
Network Operations Analyst
Network Specialist
Network Technician
Telecommunications Technician
Wide Area Network Technician

Information Services and Database Job Titles:

Data Analyst
Database Administrator
Database Analyst
Database Architect
Database Developer
Database Specialist

Programming and Software Development:

Database Software Technician
Entry (Junior Level) Programmer
Operating Systems Specialist
Programmer Analyst
Senior Level Programmer
Software Applications Specialist
Software Designer
Software/Application Support
Systems Analyst
Test Specialist

Interactive Media, Multi-media and Web Job Titles:

Animator
Audio/Visual Specialist
Graphic Designer
Graphics Technician
Imaging Specialist
Instructional Designer
Interactive Digital Media Specialist
Media Designer
Multi-Media Programmer

Multi-Media Specialist
Multimedia Technician
Production Assistant
Project Manager
Quality Assurance Technician
Virtual Reality Designer
Visual Design Consultant
Web Content Designer
Web Designer
Writer

Technical and Customer Support Job Titles:

Call Center Support Representative
Computer Operations Technician
Computer Operator
Customer Service Representative
Help Desk Technician
IS Operator/Analyst
LAN Applications Support Analyst
Lead Customer Service Coordinator
Operations Scheduler
PC Support Specialist
PC System Coordinator
PC Technician
Product Support Engineer
Software Application Specialist
Systems Administrator
Technical Sales Consultant
Technical Support Engineer
Technical Support Representative
Technical Writer

Systems Engineering and Staff Analyst Job Titles:

Computer Analyst
Development Engineer
Engineering Specialist
Member, Technical Staff
Network Test Engineer
Principal Engineer
Programmer Analyst
Project Engineer
Project Supervisor
Senior Engineer
Software Development Manager
Software Engineer
Software Quality Engineer

Software Test Engineer
Systems Analyst
Systems Engineer
System Support Engineer
System Test Engineer

While every effort was made to classify these job titles objectively, there is certainly room for titles to appear in multiple areas or areas different from where they are listed here. The primary intent for the classification here is to provide some direction as to the educational requirements for a student to take one of these positions. Hence, the belief that a position can be placed in multiple places merely indicates a need for cross-disciplinary education.

Phase 3: Cluster Definitions and the Curriculum Model

It might seem that these two things should be separated as activities and in fact, this phase was developed in two distinct parts. However, since the clusters are an integral part of curriculum model, it makes sense to keep them together here.

What follows is a template for creating a curriculum. This document does not specify a curriculum per se, but rather gives a blueprint for creating a curriculum. This provides for flexibility at many levels and for many purposes. First, it provides schools the ability to customize the presentation of a curriculum to accommodate the school's specific mission and student profile. This is useful if the school's graduates tend to serve a specific community or niche in the industry. Additionally, this flexibility allows for the evolution of curricula over time allowing for the deletion of obsolete materials and the inclusion of new ones without the need to completely rewrite the existing curriculum. The model accomplishes this by specifying that expertise be gained at specific levels within a cluster discipline to complete a curriculum rather than specifying exactly what topics comprise a curriculum. Hence, obsolete information can be replaced with new information at the same level of expertise without jeopardizing the integrity of the curriculum.

The Clusters

Clusters have been defined using the results of the "sticky notes" exercise conducted at the Provo (CITC 1) conference and suggestions from committee members. This is intended to be a living document where information can be added or altered. Each cluster title is listed (bold) with the topics for that cluster listed beneath.