Computer Science Department
Curriculum Revision – Fall 2000

prepared for

MCLA
Curriculum Committee

on

13 October 2000

Contents

Appendices

0. Introduction

Historically, the Computer Science Department's curriculum has focused on two areas: general programming and what is now being called Information Technology. Focusing on these two areas appears to have served our students well over the past decade and a half. The department has recently re-examined these two areas in an attempt to get a better handle on the state of the art in these areas.

The Computer Science Department has always attempted to prepare students to enter software positions in the computing field. We see no reason or way for this to change. Software technology, however, has changed continuously since the inception of this department. It is, therefore, the department's responsibility to insure that our program keeps up with those changes. This should be obvious if we are to take our responsibilities to our students seriously.

The members of the Computer Science Department have always tried to keep abreast of the changes in the computing industry and to keep our curriculum current. This has often required the redesign of our major curriculum to insure that we are providing our majors and minors with the knowledge and skills that will allow them to be successful in the field.

Since the completion of our last curriculum revision, existing software technologies have matured and new computing technologies have emerged. The department feels that it is once again necessary to redesign our major curriculum in order to provide the training that students in this field require. See Appendix A for a description of the motivation for developing the curriculum presented here. Appendix B contains a description of the comparison of the proposed curriculum with computing programs at other institutions conducted by the department's faculty.

A curriculum describes a list of courses and their content. To be successfully delivered, a Computer Science curriculum must specify and provide for other necessities such as hardware and software requirements; the support to install and maintain them; staffing; faculty preparation (i.e. what should the faculty be learning); etc.

The remainder of this document will present the current thinking of the department in terms of a curriculum proposal that will serve the needs of our students, along with a consideration of the other issues listed in the previous paragraph.

1. Curriculum Goals

The overall goal of the proposed curriculum will be realized via a number of sub-goals. These sub-goals can be articulated as an attempt to provide students with a strong grounding in several areas:

- Traditional computer programming,
- The design, implementation, and maintenance of large software systems,
- The design, implementation, and manipulation of databases,
- The development of multi-tiered, distributed software systems, and
- Peripheral topics related to the software technologies used to support the above (e.g. operating system concepts, data communication concepts, etc.)

It has always been the department's view that our students have been successful in the wide range of areas in which they have found employment because they were provided with a strong grounding in traditional computer programming. Our students have been well served by this and by other aspects of the department's previous curricula that, although centered on older technologies, match the areas listed above.

The proposed curriculum is not a departure from this successful formula. It is, however, a realignment of those aspects of the current curriculum which are still relevant and which have proven successful for our students with the addition of newly emerging software technologies.

Of the sub-goals listed above, the first three are an outgrowth of those successful features while the last two reflect the influence of newer technologies. Multi-tiered software systems involving distributed objects working through networks of cooperating machines have become both an interesting technology and a foundation for on-line commerce. Aspects of the curriculum related to General Programming are described in Appendix D, and a description of N-tiered Distributed Software Systems is provided in Appendix E.

2. Curriculum Model

Required:

CSCI 153 - Introduction to Programming using C++ I
CSCI 253 - Introduction to Programming using C++ II
CSCI 314 - Windows Programming
CSCI 315 - Advanced Programming using C++ I
CSCI 316 - Advanced Programming using C++ II
CSCI 319 - Object Oriented Design
CSCI 335 - Web Development I
CSCI 336 - Web Development II
CSCI 341 - Database Management & Programming I
CSCI 342 - Database Management & Programming II
CSCI 351 - Distributed Programming Concepts
CSCI 401 - Networked Systems Administration
CSCI 451 - N-tiered Software Development I
CSCI 452 - N-tiered Software Development II

Other:

CSCI 247 - Topics in Computing
CSCI 447 - Advanced Topics in Computing

3. Minor Programs

Information Technology:

Required:
- CSCI 153 - Introduction to Programming using C++ I
- CSCI 253 - Introduction to Programming using C++ II
- CSCI 314 - Windows Programming
- CSCI 335 - Web Development I
- CSCI 341 - Database Management & Programming I

One of the following:
- CSCI 336 - Web Development II
- CSCI 342 - Database Management & Programming II

Computer Science:

Required:
- CSCI 153 - Introduction to Programming using C++ I
- CSCI 253 - Introduction to Programming using C++ II
- CSCI 314 - Windows Programming
- CSCI 315 - Advanced Programming using C++ I
- CSCI 316 - Advanced Programming using C++ II

One of the following sequences:
- CSCI 335 - Web Development I and
- CSCI 336 - Web Development II

or
- CSCI 341 - Database Management & Programming I and
- CSCI 342 - Database Management & Programming II

or
- CSCI 401 - Networked Systems Administration
- CSCI 447 - Topics in Computing

Contract Minor:

With the approval of the department, a student may put together a minor program from the departmental offerings supplemented by independent study courses and/or offerings from another department. For example, minors in data communications, multi-media computing, or scientific computing might be created. Any such minor must consist of six three credit courses with at least two at the 300 level or above.

Disclaimer:

Any student completing the major course of study described previously will not be eligible to also complete a minor in this department.

4. Course Related Issues

A number of other issues have been considered which relate to the proposed curriculum. Each of these will be examined here.

Writing, Speaking, and Research:

Thus far those aspects of the proposed curriculum which directly relate to the computing components of the curriculum have been addressed. There are other skills that students majoring in IT will need to be successful in this field. These include the ability to communicate and the ability to investigate prior research in this field. The proposed curriculum therefore identifies individual courses where one or more of these important skills will be an integral part of the course. These skills will be emphasized via explicit writing assignments, research assignments, and presentations in the identified courses. The following is a list of the courses and an indication of which skills will be required in those courses:

Course	Writing	Speaking	Research
CSCI 247 - Computing Topics	X		
CSCI 319 - Object Oriented Design	X	X	
CSCI 336 - Web Development II	X	X	
CSCI 342 - Database Management & Programming II	X	X	
CSCI 351 - Distributed Programming Concepts	X		X
CSCI 401 - Networked Systems Administration	X		
CSCI 447 - Topics in Computing	X		X
CSCI 451 - N-tiered Software Development I	X		X
CSCI 452 - N-tiered Software Development II	X	X	

Mathematics:

Students completing this program will be required to complete two mathematics courses, Statistics and Discrete Mathematics. The former is currently offered by the Mathematics Department, and needs no further discussion. The Discrete Math course, however, is not currently offered by the Mathematics Department. The faculty in the Computer Science Department intend to articulate those topics that they feel are relevant for an IT major and will present that information to the Mathematics Department. If that department is unable to provide such a course it will be offered by the Computer Science Department – either as an IT course or cross listed with the Mathematics Department.

Interdisciplinary Cooperation:

In an attempt to foster interdisciplinary cooperation with other departments on campus, to take advantage of the skills and knowledge of faculty in those departments, and to provide for a broader and more complete education for majors in this department, the proposed curriculum includes a mechanism for having faculty from other departments come to various courses to ‘guest lecture’ on their area of expertise as it relates to information technology. The following is a list of those courses which the faculty feel can benefit from interdisciplinary cooperation with, and from, other faculty on campus:

Courses	Topics:
CSCI 314 - Windows Programming, CSCI 341 - Database Management & Programming I, and CSCI 335 - Web Development I	User interface design Human factors
CSCI 451 - N-Tiered Software Development I, and CSCI 452 - N-Tiered Software Development II	Human Factors Organizational Behavior Privacy Social Implications of computing Virtual Communities

5. Example Course Offering Sequence

Year One:

Fall Semester:

CSCI 153 - Introduction to Programming using C++ I

Spring Semester:

CSCI 253 - Introduction to Programming using C++ II

Year Two:

Fall Semester:

CSCI 314 - Windows Programming
CSCI 315 - Advanced Programming using C++ I

Spring Semester:

CSCI 316 - Advanced Programming using C++ II
CSCI 319 - Object Oriented Design
CSCI 341 - Database Management & Programming I

Year Three:

Fall Semester:

CSCI 335 - Web Development I
CSCI 342 - Database Management & Programming II

Spring Semester:

CSCI 336 - Web Development II
CSCI 351 - Distributed Programming Concepts

Year Four:

Fall Semester:

CSCI 401 - Networked Systems Administration I
CSCI 451 - N-tiered Software Development I

Spring Semester:

CSCI 452 - N-tiered Software Development II

Topics courses will also be offered on a regular basis as required.

6. Transfer Students

Since the curriculum will no longer consist of a list of courses that can be taken independent of each other and since it is much more integrated, courses will frequently need to be completed in a sequence. This will make it more difficult for transfers who wish to complete the program. The exact time required for the completion of the program will depend on the courses a student transfers into MCLA. For a student with a minimal number of courses (just the first two C++ courses) the program would require three years to complete. Students with additional courses to transfer in may be able to complete the program in two years.

The Admissions Office will need to assure that community colleges that tend to be feeder schools for MCLA are informed of the new program and in particular the course requirements for the first two years. With appropriate advising, students from those schools can still make a fairly smooth transition to MCLA.

7. Transition Plan

A transition plan was developed to map the phase-in of the new curriculum and the phase-out of the old curriculum over a four year period. The plan includes a schedule that indicates (a) when each new course will be introduced, (b) when courses in the current curriculum will be terminated and (c) identifies courses in the new curriculum which students can substitute for courses in the current curriculum.

The transition plan insures that students who entered the program in the fall of 2000 are able to complete the current curriculum by the spring semester of 2004. The table below shows an example course sequence that a student working under the current curriculum might complete. It includes both an example course sequence and a list of options a student will have in each semester. Courses shown as ***italic*** in column two indicate the last semester that a course from the current curriculum will be offered. Courses shown as **bold** in column three indicate a course from the new curriculum that can be substituted for a course in the current curriculum.

Students entering in Fall 2000	Example Course Sequence	Other options
Fall 2000	153	
Spring 2001	253	229, 355
Fall 2001	315	247, 261
Spring 2002	316	247, 323, **341**, 421, 447
Fall 2002	***318***	247, 447
	252	
Spring 2003	***323***	247, **341**, **351**
	447	
Fall 2003	***261***	247, **401**, 447
Spring 2004	***421***	**341**, **351**, 447

It should be noted that, with proper advising, it is possible for a student who enters the program in the fall semester of 2000 to complete the new curriculum if he or she elects to do so.

During the spring semester of 2001 currently matriculated students will be informed of the impending curriculum changes. The transition from the current curriculum will be explained to them, including information about when a course will be offered for the last time, the mapping of courses from the new curriculum which can be substituted for courses in the current curriculum, and the options for completing the new curriculum.

8. Support

Hardware:

Like Computer Science, Information Technology depends on access to equipment by both faculty and students. Unlike Computer Science, where the lack of up to date facilities can be partially excused by a legitimate interest in "timeless" as well as current computing topics, Information Technology topics and software tools are all current and require current hardware. Further, N-tiered distributed systems require multiple servers as well as workstations. Some of these servers work in a back room, are run by professionals (i.e. non-students) and support courses early in the sequence. Others need to be available to students in later courses when installation, configuration and maintenance of hosts becomes the topic of study. Given these facts, support for an initial and for a sustaining hardware budget is necessary.

The initial hardware investment will be for a data projector, a server, an uninterruptible power supply and the replacement of faculty PCs. This cost will be $18,000 - $20,000. Subsequent years will see the need for another server, for communications equipment to be used for demonstrations and for lab experiments, and for regular replacement of both faculty and lab PCs over a 3 year period. This will come to around $15,000 per year.

Software:

While hardware will be a continuous, large expense, software has proven to be relatively inexpensive due to either generous educational discounts or to free licensing for educational purposes. Although staying up to date is again a critical need, a continuing budget of $3,000 should take care of software needs.

Personnel:

The combination of hardware and software above is beyond the ability of the current Computer Support Services staff to maintain for the department. A staff person assigned to the department – similar to one who might maintain laboratory equipment for a department such as biology or physics – could solve the problem. A second option would be a person assigned to Computer Support Services with the clear understanding that IT needs should be his or her priority. Alternately, department faculty could be assigned alternative service to support the hardware and software as they have done in the past.

9. Staffing (Faculty)

A three and one half member faculty will be just able to cover the courses in the proposed curriculum. Projections of Core Curriculum demands call for an average of seven courses per

semester in service. Non-Core service courses, department members volunteering for duty in Tier 1 Science & Technology courses, hardware/software maintenance as alternate service, and on-going curriculum development will increase this to require at least six full-time faculty to maintain the program. These numbers are based on current admission figures and may have to be reconsidered in the future if enrollment continues to increase (incoming numbers in CS were up by 150+% this fall).

10. Advisory Board

The department feels that it would be beneficial to constitute an advisory board made up of individuals from the types of industries that students completing this program will find themselves. The intent is to identify approximately six to nine individuals who are willing to serve in this capacity. The department faculty would arrange to meet with this group twice a year to keep abreast of the state of the industry.

Appendices

A. Motivation for Curriculum Revision

Throughout the history of the Computer Science Department there has been a continual evolution in the department's major and minor programs. These changes have resulted in courses being added, removed, or modified in an attempt to fine tune a curriculum. However, this evolution has periodically been punctuated by changes which were significant enough to require a major curriculum revision. This is to be expected in a field that has changed as quickly as computing has over the past few decades.

Recently evolutionary changes in the department's curriculum were made to fine tune the curriculum in an attempt to keep it in sync with the changes in this field, changes in the computing industry, or to better serve our students' needs. The course CSCI 447, Topics in Computer Science, has been used as a mechanism to 'field test' new technologies to evaluate them in terms of eventually integrating them into the department's core curriculum. The most recent transition of technologies from the realm of Topics courses to the department's core curriculum was Object Oriented Programming and the C++ programming language. The former is an approach to program development, and C++ is a vehicle for teaching that approach.

However, once again it is time for a major revision in the Computer Science Curriculum. The motivations to make various major program revisions to the Computer Science Curriculum have come from both internal and external forces. Internally the department has always attempted to be sensitive to the needs of its students in terms of providing them with a sound education in computing which would prepare them to enter the computing field as professionals. The success of this attempt can be measured by the success of students who have completed a major in computer science, and have taken their places in the computing field.

The external forces that have motivate the department to revise its curriculum in the past include the demands of the computing industry in terms of the preparation of students to successfully participate in the computing industry as professionals; program reviews conducted by the Board of Higher Education; and philosophical adjustments to the mission of the college.

The current curriculum revision was motivated by both internal and external forces. Internally, the members of the department were examining what changes might best help our students prepare for their futures in the computing industry given the significant changes in that industry over the past few years.

This examination was intensified after careful examination of an external document from the Board of Higher Education which addressed the current state of computing in the commonwealth, a preliminary review of all computing programs in the state system, recommendations for computer science education in general, and for MCLA in particular. The cover letter to this document notes "… there is much work to be done by all of us to improve the quality of IT programs and the supply of trained IT workers." (IT here stands for Information Technology.) The importance for both employers and potential employees in terms of preparing qualified IT workers is further emphasized in the introduction to this document. Citing a report by the Department of Commerce it notes "… it claims that Massachusetts will annually have an average of 5000 unfilled information technology jobs between 1999 and 2006; jobs that could pay an average of 78% more income than non-information technology jobs …"

The report points out clearly that in terms of the baccalaureate level computing programs in the commonwealth that "Four-year programs have two purposes, to prepare graduates to enter the

workforce or to continue on for graduate studies. Most of the students will enter the workforce after completion of their undergraduate studies, though a significant number of these students will return for at least some graduate work, either to qualify for a better position in the profession or to keep up to date in this rapidly changing field."

The report suggested that all Computer Science programs be accredited by the Computer Science Accreditation Board (CSAB) within the next five years. However, in the short section addressing the preliminary evaluation of the Computer Science program at MCLA, the report concludes that "The degree program might be more suited to computer applications (a.k.a. information technology) rather than computer science."

The evaluators went on to point out what the faculty in this department have known for almost a decade, this department and this college will never be able to compete in terms of the standards set by the Computer Science Accreditation Board (CSAB) for Computer Science programs. Accreditation by CSAB is better suited to programs and institutions with the resources and mission to prepare students using more of an engineering model. The emphasis on faculty research, an extremely large number of required credits in the degree program, a large number of credits in Mathematics, a large number of credits in science, etc. all make such a program impractical at most small four year institutions, let along one who's mission is to both prepare students for employment and to provide them with a Liberal Arts education.

The faculty spent some time identifying the strengths and weaknesses of the department and the college in terms of the preparation of graduates who will be successful in their chosen area of study. The following is a brief articulation of the department's conclusions in terms of its history of preparing students for the computing field:

- the department has always attempted to prepare students with the knowledge and skills for them to enter the software end of the computing field
- there is no reason for this to change
- software technology, however, has changed since the inception of this department, and continues to change
- it is the responsibility of the department to insure that the program offered keeps up with those changes

The Computer Science Department faculty concluded that historically its curriculum has focused on two areas: general programming and what is now being called Information Technology. Focusing on these two areas appears to have served our students well over the past decade and a half. The department has recently re-examined these two areas in an attempt to get a better handle on the state of the art in these areas.

The department has always attempted to prepare students to enter software positions in the computing field. We see no reason or way for this to change. Software technology, however, has changed continuously since the inception of this department. It is, therefore, the department's responsibility to insure that our program keeps up with those changes. This should be obvious if we are to take our responsibilities to our students seriously.

B. Comparison with Other Institutions

In addition to the document from the Board of Higher Education, and the department faculty's evaluation of the computing field, the department conducted an examination of a number programs offered by institutions identified as peers with MCLA, and by state colleges in Massachusetts. This examination was conducted to determine what those institutions were offering, and to allow us to compare the proposed curriculum with those other programs.

All of the peer institutions and our sister institutions in Massachusetts offer computing majors. At some schools, a stand-alone Computer Science department offers these majors while, at others a Mathematical Sciences department does. In one school the Natural Science division offers a program.

In the majority of cases these majors resemble our current Computer Science major to one degree or another with the exception that several of the programs of study tend more towards the theoretical than does ours. In any case, all are based on generally accepted course topics that are selected from curricula published by various standards bodies, notably by the Association for Computing Machinery or the Computing Sciences Accreditation Board. No program seems to have attained accreditation.

No departments in the selected institutions seem to offer a modern Information Technology program. One institution does offer an Information Systems option within the Computer Science major.

The program that was most closely aligned with our curriculum proposal was not directly in a peer institution. Regis University (as opposed to Regis College) has several tracks at both the undergraduate and the graduate level through its School for Professional Studies. One track, the Master of Science in Computer Information Systems in E-Commerce, fits well with what we propose to do at the undergraduate level. Although no details were available on line, the School for Professional Studies also offers undergraduate tracks in Computer Information Systems and Computer Networking.

Since none of the institutions identified above had programs in Information Technology a search was conducted of other institutions which did offer such programs. This search resulted in a number of institutions, both here in the United States and abroad, which offer Information Technology programs.

Close to home Umass Lowell offers a certificate program in Intranet Development. This program includes courses in Application Development, LAN/WAN Technologies, Relational Database Concepts, C++, Java, etc. Though the titles may differ, the content of these courses is similar to those proposed here. These include Web Development, Database Management & Programming, and Networked System Administration.

Bond University identifies itself as "a small private university in Australia consisting of five academic units: the schools of Business, Humanities & Social Sciences, Information Technology, Law and the Institute of Health Sciences. The University is situated on a beautiful campus on the Gold Coast, about one hour drive away from Brisbane, the capital of Queensland." A review of the Information Technology program offered found Software Development, Database Management, Internet Technology, Network Computing, etc. as required

courses for all students, and then a concentration in one of a number of 'Discipline Majors'. One of those Discipline Majors is Internet Technology. This major includes courses in Desktop Application Development, World Wide Web Technology, Designing Database Systems, Programming Concepts, and a project course. This program is very similar to the curriculum being proposed here.

Tarrant County College (South Campus) is a two-year, comprehensive community college located just inside the Southeastern portion of Loop 820 in Fort Worth, Texas. Although this a community college it was of interest because it offered two degree programs, one in Network Technologies, and one in Web Development. Examining the requirements from both of these programs found that the curriculum being proposed here contains courses from both of these degree programs. This is what one would expect considering that the proposed curriculum leads to a bachelors degree. The courses in the proposed curriculum are offered at a higher level, and many are two semester sequences as opposed to those offered at Tarrant which are one semester courses. However, the topic list is very similar to that of the proposed curriculum.

American Intercontinental University offers a Bachelor of Information Technology degree which is very similar to the one being proposed here. It includes courses in general computer programming (e.g. Principles of Computer Programming, Introduction to Data Structures). Two courses are required, Introduction to Computer Networks and Computer Software Concepts, Installation, Configuration & Support, which map to the Networked System Administration course in the proposed curriculum. There are also required courses in Database Applications, Visual Basic Programming, and a capstone experience.

Dominican University, in River Forest, Illinois, offers an undergraduate degree that is called "Computer Information Systems". This degree program requires students to complete courses in general computer programming (C++, Object Oriented Programming, etc.) which is consistent with the general programming component of the curriculum being proposed here except that where Dominican requires only two semester of general programming, the proposed curriculum requires four. The other courses that make up the Dominican program include Windows Based Application Development, Database Organization & Design, LAN Administration, and an internship. The program being proposed includes courses in all of these areas except for the internship, and, again, the proposed curriculum requires a two semester course sequence in some of these areas. The proposed curriculum does not require an internship, but does include a two semester projects course that will require students to build a complete N-tiered Distributed Software System (see Appendix D for a description of this type of system).

From this review of programs it was concluded that although there are currently no state colleges in Massachusetts that are currently offering a degree program in Information Technology, and that none of the institutions identified as MCLA's peers are offering this type of program, there are other colleges and universities that offer programs that are consistent with the one being proposed for MCLA.

C. Outside Evaluation

Before proceeding too far in the direction that the department was considering for its curriculum an early draft of the proposed curriculum was sent out to Bryan M. Blair, Chief Information Officer for The Hartford Companies, for his comments and suggestions. Bryan provided an in depth critique of the draft which was very helpful in fine tuning the curriculum proposal. He

confirmed the faculty's position that a strong general programming component was important in preparing students in this field, and that Object Oriented Programming and C++ were the appropriate means for doing so.

In terms of the Information Technology part of the curriculum Bryan was supportive of the overall structure of this component, and had a number of suggestions for particular technologies to be introduced in the courses in this area. Some of his suggestions found their way into later drafts of the curriculum. Other suggestions made by Bryan were not considered appropriate for MCLA for one of two reasons: (a) they were too specific to the actual needs of The Hartford Companies, and (b) due to budgetary consideration the faculty decided that there were more financially reasonable ways to provide the experiences intended in the proposed curriculum.

Overall Bryan's comments were supportive, and very useful. They reinforced the fact that the faculty was interpreting the trends in the computing industry correctly, and that those trends were being reflected in the proposed curriculum.

Two members of the department met with two employees at Tripod (one was a lead engineer and the other a software engineer) to discuss a later draft of the curriculum. Again, the Tripod folks were very supportive of the curriculum, and made very useful suggestions for strengthening it. On their recommendation the Java programming language was given more coverage in the curriculum than originally thought necessary.

A copy of the final draft of the proposed curriculum was sent out to a number of people in the industry. These included management personnel at General Dynamics Defense Systems Techsight, a software engineer with GDDS who is involved with developing systems using object oriented technologies, a Database Administrator at CSC, and an engineer at Mindbranch here in North Adams.

Thus far only a few responses have been received from these requests for comments on the proposed curriculum (see below). It was decided to submit the proposal based on the feedback from those individuals and on the feedback received on the earlier drafts of the proposal since there have been no significant changes made to the proposal.

Rich Garabedian, an engineer at Mindbranch, began his comments with: "… overall I like the program quite a bit. It addresses many needs that I see developing in our "local workplace". I believe these needs are not just local either, they would apply to anyplace across the nation." He did point out that there was no specific course identified which addresses 'design'.

Rich concluded his assessment of the curriculum with the following: "I wish I had gone through this curriculum. It would of positioned me much better with regards to the kind of work I do today."

James Webb, Software Engineer at General Dynamics Defense Systems, also submitted comments. He was very supportive of the program. However, he also pointed out that it would be important to maintain the topics from Object Oriented Concepts (a course in the current curriculum) as part of the new curriculum. This course focuses on object oriented design, as well as some other related topics.

A request for comments was sent to management personnel at General Dynamics TechSight . We were informed that those managers put together a working group to evaluate the draft of curriculum proposal we sent to them. Informally we have heard from that working group. Again, the primary issue raised was the lack of an identifiable course in object oriented design.

Initially the faculty had identified two places within courses in the curriculum where design would be addressed formally. This topic could be introduced in the Database Management and Programming II course and could be further examined and applied in the N-tiered Software Development course sequence. However, given the comments received by the reviewers it has been decided to modify the original proposal to include a separate course which addresses design issues from an object oriented perspective. CSCI 319, Object Oriented Design is that course.

Additional comments from those asked to review the proposed curriculum will be made available as an addendum when they are received. For now the department feels that the proposal has been accepted as meeting the needs of the students, and those of the industry, based on the comments about the curriculum received thus far.

D. General Computer Programming

In terms of general programming it is clear that Object Oriented Programming (OOP) is the dominant model used in the computing field today. The C++ Programming Language provides a mechanism for teaching both computer programming and OOP and is consistent with the demands of industry for competent C++ programmers.

The proposed curriculum includes a four course sequence in C++ and OOP. This is a reduction from the requirements in the current curriculum where there are five courses addressing these topics directly. Therefore, the topics covered in those five courses, and others not currently covered, need to be mapped into the four courses in the proposed curriculum. It should be obvious that the amount of material covered in five courses, plus the material that is not currently covered in those five courses but that is becoming more important, can not be done justice in a four course sequence without careful evaluation of what are the most critical topics, and including only those topics in the course sequence. The critical topics to be covered include simple classes, inheritance, template classes, virtual classes, polymorphism, interface programming, exception handling, run time type identification (RTTI), multiple inheritance, and the standard template library.

Although this section addresses the general programming sequence in the curriculum, it should be noted that Object Oriented Programming will be the working model throughout the curriculum. Other programming languages that either support this model directly (Java), or are object based (Visual Basic) are used in courses throughout the curriculum.

E. N-tiered Distributed Software Systems

As noted above, the proposed curriculum includes a strong computer programming sequence based on an Object Oriented Programming model. In addition the curriculum includes a sequence of courses designed to expose our students to the development of N-tiered distributed software systems.

A closer look at the ‘N-tiered distributed software system’ model will provide a better understanding of what this model entails. The N-tiers refer to the decomposition of a software system into a number of pieces which must communicate with each other and work cooperatively. For example, an interface may be developed which presents information to a user via the World Wide Web. The web browser sends requests for information to a web server. The web server then passes it to a middle level of software. The middle tier contains the logic to locate the desired information by retrieving it from ‘back end’ servers, usually database management systems. That information is then put into a form that can be sent back to the end user as a web page via the web server.

One of the characteristics of this type of system is that the components do not necessarily form a dedicated system, but can be constructed more as a cooperating system. That is, the web browser is not a dedicated part of this system and can be used to browse other material from the Internet. The middle tier is not a dedicated part of the system, and can be queried by different means including a web browser and also by more traditional applications written in programming languages like Visual Basic and C++. The ‘back end’ database management system is not a dedicated part of this system, and can therefore be used by other applications.

To understand this model students will need to understand the mechanisms that support it, along with the more visible technologies. That is, in addition to an understanding of how to build interactive web pages, interactive applications, middle tier logic, and back end databases, a student will need to understand how the mechanisms that support such a system work, how they are integrated into the various operating systems, how those individual systems are networked together, how to access software components that are distributed throughout the network, and how to build software components that can be distributed throughout the network.

The current curriculum is based primarily on the design and implementation of ‘stand alone’ software systems. That is, software systems that support some application, but which are self contained in the sense that they run on one computer, and allow for the interaction with one or more users. That is not to say that other current technologies are ignored, but they are covered in discrete courses addressing the individual technologies – operating systems, visual programming, networking, data communication, web page design, system analysis, database management, etc.

The proposed curriculum attempts to realign many of these technologies, along with additional ones, to provide for a more integrated sequence of courses - a sequence of courses which are interdisciplinary within the field of computing.