# Diary of a CS/IS Collaboration: Responding to Computing Curricula 2001

Robert Nehmer[1]
Dept. of Accounting, Finance, and Information Systems, Berry College
Mount Berry, GA 30149

and

Kris Powers[2]
Dept. of Mathematical Sciences, Berry College
Mount Berry, GA 30149

## Abstract

In this paper, we describe our ongoing efforts to update our computer science and information systems majors. We consider both the implications of recent curricula recommendations for each individual program, as well as for the joint coordination of the two programs. While our efforts are inherently tied to our particular institutional context, we hope that they might serve as a model for similar efforts elsewhere.

**Keywords:** curriculum design, shared computing core, CS/IS curriculum

## 1. Introduction

The recommendations of *Computing Curricula 2001* are intended to be published in a number of volumes. The first of these was the *Computer Science* volume (*CC01-CS*) that was published in the fall of 2001. The *Information Systems* volume (*CC01-IS*) is expected in the near future (Davis, 2001). Once all of the discipline specific volumes have been completed, an overview volume of *Computing Curriculum 2001* "… will contain definitions of the various computing disciplines along with an assessment of the commonalities that exist in the particular approaches." (*Computing Curricula 2001*, p. 1) Because it will follow all of the other reports, production of such a volume is not anticipated in the near future.

In this paper, we describe our ongoing efforts to update our computer science (CSC) and information systems (IS) majors. Our efforts are not only a response to the recommendations of *Computing Curricula 2001*, but also mirror the structure of those reports. In the last academic year, we initiated a much-needed update of our CSC curriculum that significantly altered the content and pedagogical approach of this major. The update was based largely upon *CC01-CS*. In response to that effort, we are now in the process of re-assessing our IS major. We had anticipated that *CC01-IS* would serve as a primary reference for this work. While this volume has yet to be published, initial reports from Task Force members have been presented. One such report indicates that substantial changes to *Information Systems* '97 (*IS97)* are not expected (Davis, 2002). Therefore, we have been looking to the older *IS97* guidelines thus far. Finally, we are also examining the interplay between our CSC and IS majors, and evaluating the potential for sharing resources between these programs. While our efforts are inherently tied to our particular institutional context, we hope that they might serve as a model for similar efforts elsewhere.

## 2. Background

### Institution

Berry College is a small, comprehensive, liberal arts college. Computing-related majors are offered from two different schools within our institution. The Department of Mathematical Sciences is housed in the School of Mathematical and Natural Sciences. As is

[1] rnehmer@berry.edu
[2] kpowers@berry.edu

common at many small colleges, this is a combined department with both computer science and mathematics faculty. It offers degrees in mathematics, mathematics education, computer science, and, in conjunction with our School of Business, a degree in decision science. The newly revised requirements for a major in computer science are listed in Appendix A, and those for decision science appear in Appendix C. The School of Business is home to the Department of Accounting, Finance, and Information Systems; this Department supports a business administration degree with concentrations in accounting, finance, and information systems. Recently, the School of Business approved a conversion of the concentrations into full majors. The current requirements for a concentration in information systems (based on the old CSC curricula) are listed in Appendix B. The students of all three programs have participated in research projects with faculty including database web applications, examining the eXtensible Business Reporting Language (XBRL), and distributed computing applications.

**Computing Curriculum 2001**

**Computer Science:** The *CC01-CS* curriculum identifies 14 different knowledge areas for the CSC discipline and a collection of core topics within each of these areas. These core topics are intended to represent the minimal subset essential to any viable CSC curriculum; they are not intended as a complete curriculum. A CSC program is viewed at three different levels of coursework: introductory, intermediate, and advanced. Core topics are generally covered at the lower two levels, while advanced topics are generally unique to the institution. Of particular note are the models for the lower two levels.

Most traditional for the introductory sequence is the *programming-first* model, with various flavors possible depending upon the paradigm of the chosen language (procedural, object-oriented, functional, or logical). Also well known is the *breadth-first* model, first proposed in *Computing Curriculum 1991*. This model offers students a broad exposure to the CSC discipline early in their academic careers. While this approach seems to offer great potential for improved learning, that potential has gone largely unrealized. At most institutions that have attempted this model, it consists of a single, initial breadth-first course, followed by the traditional programming sequence. This approach is most often criticized for delaying development of student design skills.

At the intermediate level, most institutions still follow the typical topical approach, in which each knowledge area is taught in isolation. The *CC01-CS* report urges the adoption of new models that cut across knowledge area boundaries. Course work in such models would focus on broad themes that transcend the topical boundaries.

**Information Systems:** The *IS97* Model Curriculum consists of 127 learning units loaded into 10 courses. The courses are further categorized into three levels and five presentation areas. The three levels are General, Major and Minor, and Major. The General level consists of material relevant to all business students. The Major and Minor level contains material relevant to IS majors and IS minors. The Major level is comprised of material specific to the IS major. The five presentation areas are IS Fundamentals, IS Theory and Practice, Information Technology, IS Development, and IS Deployment and Management Processes. The first two areas are considered to be at the General level. The third area and part of IS Development are relevant to both majors and minors. The remainder of IS development and IS Deployment and Management Processes are at the Major level. The course sequencing in the program is generally seen as proceeding from level 1 to level 2 and then to level 3.

The courses at level 1 are Fundamentals of IS, Personal Productivity with IS Technology, and IS Theory and Practice. These courses are typically underrepresented in business school curricula. A typical curriculum consists of either one or two required IS courses. Usually a single course would be a combination of the first two level 1 courses listed, although many single courses are nearer the second. IS Theory and Practice might be added to a curriculum as the second required IS course for a business degree. *IS97* is, therefore, heavy in the fundamentals and theory areas when compared to most business school offerings. The *CC01-IS* proposal is even heavier since it suggests the Personal Productivity course be relegated to prerequisite status and adds an electronic business strategy course to level 1.

At levels 2 and 3, the *IS97* suggests three courses each in the Information Technology and IS Development areas. There is then a capstone course in project management. These two levels are relatively unchanged in the *CC01-IS* proposal. It is in the Information Technology area and the level 1 coursework that problems develop when the IS curriculum is articulated with a CSC curriculum. This is because the CSC introductory sequence is a closer fit to the Information Technology area than to any other area in the IS curriculum. Yet, the *IS97* structure has level 1 courses as prerequisites to the Information Technology area. This implies an inversion of level 1 and the IT area in a shared program.

## 3. Curriculum

**Computer Science**

Prior to the fall of 2001, our degree program in CSC had been maintained by three faculty members with modest backgrounds in the discipline. Further, none of these faculty members was assigned to teach in the program full time. Unsurprisingly then, the CSC curriculum had fallen into a sad state of disrepair, and was in need of serious revision. In the fall of 2001, a major revision effort was spurred by the addition of a new CSC faculty member, and by the steelman draft of

*CC01-CS* (although other curricula recommendations were also considered – e.g., Walker, 1996).

The resulting degree program is summarized in Appendix A. This new curriculum represents a significant change in the content and pedagogical approach of this major. While a complete discussion of the rationale for this curriculum is beyond the scope of this paper (see Powers, 2002a for a complete discussion of the curriculum, and Powers 2002b for a description of the introductory sequence specifically), several salient aspects should be noted.

As is discussed in *CC01-CS*, the content of the traditional two-course introductory sequence has changed over time to include more topics of ever-increasing complexity. For example, a solid foundation in programming and design using a modern object-oriented language (as opposed to using an out-dated procedural language) requires newer topics like inheritance and polymorphism. Because full coverage in two semesters has become increasingly difficult, the *CC01-CS* Task Force strongly endorsed the idea of moving to a three-course introductory sequence. The extra time allows full coverage of a modern object-oriented programming language and system design. However, while extra time for these topics is clearly needed, an entire course is arguably excessive. The *CC01-CS* Task Force suggests that some of the time available in a three-course introductory sequence might be effectively used to incorporate additional, less usual topics.

We considered such an approach as especially attractive, as it holds the potential to exploit the "spacing effect." This theory of educational psychology holds that learning is improved by increasing the time frame over which the learning task is attempted. The spacing effect is "one of the most dependable and replicable phenomena in experimental psychology... remarkably robust ...(and) truly ubiquitous in scope." (Dempster, 1988) Based on educational psychology studies, the "spacing effect would seem to have considerable potential for improving classroom learning, yet there is no evidence of widespread application." (Dempster, 1988) We decided to try to realize this potential by spacing our coverage of design throughout what eventually evolved into a four-course sequence for the first two years: *CSC 120-121 Principles of Computer Science I & II, CSC 220 Data Structures and Multilevel Machines*, and *CSC 320 Algorithms and Models of Computation*. These courses cover the topics usually found in the CS1 – CS2 introductory sequence (i.e., fundamentals of programming and data structures) , as well as about the first half of CS3 (i.e., algorithms). The remaining course time is used in several ways.

Like the programming and design topics, the collection of discrete mathematics topics recommended in *CC01-CS* also exceeds what can be discussed in a single course. One possible solution is to expand the discrete mathematics requirement to a two-course sequence. However, in our context this option was considered neither feasible nor desirable. Limited resources prohibit the addition of a second discrete mathematics course geared for CSC majors; we simply don't have a sufficient number of majors to support it. In addition, the transference of the problem solving skills learned in mathematics courses to CSC applications is being increasingly questioned. (see, e.g., Robertson, 2000) And, finally, many are expressing rising concerns that CSC students see mathematics as divorced from, or at best peripheral to, their CSC studies (see e.g., *Working Group on Integrating Mathematical Reasoning into Computer Science Curricula)*. Therefore, we chose to use some of the remaining course time in our new sequence to allow additional discrete mathematics topics to be taught in their application context.

The primary use of the extra course time, however, is to allow the incorporation of breadth topics. This combination of the programming-first and breadth-first approaches might be most aptly termed "breadth-also." This alternative to breadth-first retains the advantage of providing an early exposure to the breadth of the discipline, but avoids the disadvantage of postponing a student's introduction to design. A critical feature here is that the breadth topics include the full range of problem solving processes for CSC discipline (as identified in *CC91*): mathematical theory, experimental abstraction, and engineering design.

Another facet of our introductory sequence is that it includes a closed laboratory component to provide an environment for collaborative problem solving and experiential learning. The evolution of computer science into a laboratory-based science is recognized in *CC01-CS*. Moreover, while pragmatic issues may make closed laboratories difficult at large institutions, curricula guidelines for small, LAS model institutions require closed laboratories for introductory CSC course work, and even recommend further laboratory based courses. (Walker, 1996) It is notable that at our institution the addition of this laboratory time makes each course in the sequence a 4-hour rather than a 3-hour course.

Finally, it should also be noted that the design of our new CSC curriculum did not ignore the implications for IS. Recommendations outlined in both *An Information Systems-Centric Curriculum '99* (*ISCC99*) and *Information Systems '97* (*IS97*) were reviewed and incorporated. *ISCC99* claims that when their curriculum is implemented based on an existing computer science program it uses courses in: CS1, CS2, Computer Organization, File Structures, Database, Human-Computer Interaction, Telecommunication and Networking. Their curriculum also includes discrete mathematics. Similarly in *IS'97*, topics in the IS-CS overlap are found in the Information Technology area, as well as in some of the level 1 material. Finally, we also considered which topics were included in the current IS program. In general, courses that address these areas were retained in our curriculum, and in some cases moved to a lower level where they might be more accessible to IS majors.

*ISC99* cites a number of problems with building a program this way (i.e., using coursework in CSC), including a lack of both early experience with collaborative work, as well as (an early) focus on system design and implementation. The recommendations of *IS97* seem to agree: "[This curriculum] responds to industry requests for both increased emphasis in technical orientation and improved skill in individual and group interactions." We have attempted to design the new CSC curriculum in a way that avoids these problems. The second course in our new introductory CSC sequence (*CSC 121Principles of Computer Science II*) includes a group project to design and implement a system. To support this educational experience, earlier laboratory work in the first course (*CSC 120 Principles of Computer Science I*) incorporates initial collaborative work experiences.

**Information Systems**

The IS concentration in the School of Business is three years old. Its first alumni graduated in 2001, largely from a group of students who changed from the CSC to the IS program after the latter was implemented. It is unique to the School in that the typical concentration requires 15 to 18 hours beyond the business precore and core, while the IS concentration requires 24 hours.

The curriculum redesign in IS is being justified as a reaction to the curriculum changes in the CSC curriculum, and is therefore mainly concerned with the 12 hours of computer science requirements in the IS concentration (see Appendix B). Of the CSC courses included in this concentration, only *CSC 404 Organization of Programming Languages* has been retained in the new CSC curricula in its original form. *MAT 317 Discrete Mathematics* has been replaced by a more applied course *MAT 219 Discrete Structures*. Material from the other CSC courses in the IS concentration has been reorganized into the new CSC introductory sequence (*CSC 120-121-220*), and the two upper level courses: *CSC 320 Algorithms and Models of Computation* and *CSC 350 Computer Organization and Architecture*. Finally, it should be noted that the new IS curriculum will be a major. However, the move from concentration to major is largely a cosmetic one.

Experience in teaching the IS requirements for the past 2 to 3 years in the IS program has led the IS faculty to have the following requirements of the CSC background of the IS students. IS students need a strong systems foundation including concepts of modularity, hierarchy, interfaces and communication, and scale. Good problem solving and analytical skills are also necessary in addition to a strong foundation in several languages, preferably Visual Basic and Java. Finally, an appreciation of the mathematical foundations of the systems sciences is required. The problem facing the IS faculty is how to replace the 12 hours of CSC coursework while fulfilling these requirements and meeting the College's resource sharing objectives.

**Decision Science**

The Decision Science (DS) program is a shared responsibility between the School of Business and the Department of Mathematics and Computer Science. The program has a long history at Berry College, and at graduation, its students have been some of the most suceessful placements of the College. The current curriculum design includes a three-part core (business, mathematics, and computer science) and a further area of concentration within one of those three parts (see Appendix C). The core course offerings in computer science are the most germane part of the curriculum to this paper. They have already been adjusted to follow the new curriculum offerings in computer science. Therefore, we expect this to be the least controversial area in the core integration area. However, see Savage (1995) for an argument to design the mathematical part of the curriculum as well.

## 4. A Shared Core?

*IS97* discusses the relationship between CSC and IS as follows: "There is a close relationship between IS and CS. In some schools, students in both areas may take common courses. However, IS is unique in that its context is an organization and its information systems. This leads to important differences with CS in the context of the work to be performed, the type of problems to be solved, the types of systems to be designed and managed, and the way the technology is employed. … IS and CS are distinct areas of study, but they both require a common subset of technical knowledge." (*Information Systems '97*, p.7)

In May of 2002, an ad hoc committee on IS/CSC programs was formed by our College's Provost to examine how this relationship should and could be realized in our particular institution. More specifically, this committee has been charged to "...research fully the potential for sharing faculty and course resources between the IS and CSC programs, and to determine what separate resources are necessary/desirable and sustainable." The rational for exploration of resource sharing is threefold. First, there is a general desire to contain the costs of instruction while maintaining the quality of our programs. Second, neither of the programs is particularly large, and maintaining a critical mass of students in each cohort is important, as class sizes have sometimes fallen to fewer than ten. This is of special concern in the relatively new IS program. Resource sharing is seen as a way of achieving critical mass in these classes. Finally, we were intrigued by the possibilities of having theoretical (deductive) learning styles interacting with practical (inductive) learning styles in the classroom. We plan to assess the results of this collaboration with regard to the students' and their employers' perceived quality of instruction and learning.

Preliminary discussion led to a structuring of the problem domain in the following terms. There are two departments involved and three degree programs: computer science, decision science, and information systems. There is a need to determine the basic elements of each curriculum including the social dimension (group work, problem contexts, etc.). The determination of these basic elements must be driven by the objectives of the degree programs, including the type of career within the fields. It has been noted that the core liberal arts experience can produce a centered, self-directed graduate and that this could be marketed to potential employers. An architecture of the common elements of the three programs is needed, with emphasis on the commonalities of the CSC and IS programs. The development of this architecture will be informed by using the *IS97* (*CC01-IS*, if available) and *CC01-CS* as benchmarks.

Development of objectives for each of the degree programs will facilitate the development of assessment systems for the curricula. The School of Business is in the middle of AACSB accreditation, which requires all degree programs to state objectives through mission and vision statements, and to assess how well those objectives are being met. Development of objectives and assessment measures on a course-by-course or common core level will allow for preliminary assessment of the effectiveness of the new programs. These items are also included in the problem domain as we have defined it. Finally, the College's mission is to educate the head, heart, and hands. It has initiated a program to link its historic work-study approach more closely with the degree programs. That is one reason why a work experience component appears in the new CSC curriculum. The particular ways in which a student can satisfy the experience need to be further explored. The work experience element must also be included in the IS and DS programs.

Over the Summer of 2002, a provisional IS curriculum was developed by the joint CSC/IS committee. The curricular proposal is provisional because it is driven by the timing of the College's catalog cycle and not strictly by the needs of either curriculum. The proposal (see Appendix D) incorporates the two beginning level CSC courses, although a name change to *Principles of Computing* has been suggested. A discrete mathematics course is also required. On the business side, IS students would not take the general business applications (BUS 107) or theory (BUS 316) courses. Instead, two new courses are proposed to fulfill the level 1 *IS97* requirements. The rational for doing this is quite simple: IS students have already had a year of computing principles before these two courses and they will be able to perform at levels significantly above the general business students. This structure also allows the IS program to make adjustments for the inversion of the level 1 and Information Technology area mentioned earlier. The level 2 and level 3 courses remain largely the same as in the current program.

The proposed program could also be employed to strengthen the CSC program. Currently, breadth topics important to IS, like database and GUI design, are incorporated in the *CSC 121 Principles of Computer Science II* course. It is possible that instead, the IS 230 course would be required of CSC majors, and so these topics could be removed from the CSC sequence. This would free time for enhanced coverage of the remaining topics.

**5. Conclusion**

While CSC and IS have grown into rich, distinct disciplines, there still exists a great deal of commonality between them. In a small college environment like ours, it is imperative that we exploit those commonalities to provide the highest quality technologically related programs of study that we can while conserving limited college resources. Of course, the resources and limitations that guide the process of curriculum revision will vary from institution to institution. However, we hope that our efforts might serve as a helpful model for other institutions undertaking similar curricular revision.

**6. References**


*Computing Curricula 1991*, Report of the ACM/IEEE-CS Joint Task Force, ACM Press, 1991.

*Computing Curricula 2001: Computer Science*, The Joint Task Force on Computing Curricula: IEEE Computer Society and Association for Computing Machinery (ACM), 2001. Available at: http://info.acm.org/education/curricula.html.

Davis, G., D. Feinstein, J. Gorgone, H. Longnecker, and J. Valacich, "IS 20001: Updating IS '97, A Progress Report on Undergraduate IS Curriculum Development," Americas Conference on Information Systems (ACIS), panel presentation, Boston, 2001. Available at: http://info.acm.org/education/curricula.html

Dempster, F., The Spacing Effect: A Case Study in the Failure to Apply the Results of Psychological Research, *American Psychologist*, 43(1988), 627-634.

*Information Systems '97: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*, Association for Computing Machinery (ACM), Association for Information Systems (AIS), and Association of Information Technology Professionals (AITP) (formerly DPMA), 1997. Available at: http://info.acm.org/education/curricula.html

*An Information Systems-Centric Curriculum '99: Program Guidelines For Educating The Next Generation Of Information Systems Specialists, In Collaboration With Industry*, July 1999. Available at: http://www.iscc.unomaha.edu

Powers, K., "Building On Cornerstones: Implementing Computing Curricula 2001 in a Small College Context," manuscript, 2002.

Powers, K., "Breadth-Also: A Rationale and Implementation," manuscript, 2002.

Robertson, Ian, Imitative Problem Solving: Why Transfer of Learning Often Fails to Occur, *Instructional Science*, 28(2000), 263-289.

Savage, Sam L., "The DNA of Decision Science," in *Education in a Research University*, Richard Cottle, Ingram Olkin, eds., Stanford University Press, 1995.

Walker, Henry A. and G. Michael Schneider, "A Revised Model Curriculum for a Liberal Arts Degree in Computer Science," *Communications of the ACM*, Vol. 39, No. 12 (December 1996), p. 85-95.

*Working Group on Integrating Mathematical Reasoning into Computer Science Curricula*. Available at: http://www.math-in-cs.org.

**Appendix A. Computer Science Major Requirements**

***Note:*** The credit hours for each course is listed as *X-Y-Z* where *X* denotes the lecture contact hours per week, *Y* denotes the laboratory contact hours per week , and *Z* denotes the total student credit hours for the course.

***Required courses***

*Core* (28 hours)

| | |
|---|---|
| CSC 120 Principles of Computer Science I | 3-2-4 |
| CSC 121 Principles of Computer Science II | 3-2-4 |
| CSC 219 Discrete Structures | 3-0-3 |
| CSC 220 Data Structures & Multilevel Machines | 3-2-4 |
| CSC 300 Professional & Social Contexts I | 1-0-1 |
| CSC 320 Algorithms& Models of Computation | 3-2-4 |
| CSC 350 Computer Organization & Architecture | 3-2-4 |
| CSC 400 Professional &Social Contexts II | 1-0-1 |
| CSC 490WI Senior Project 3-0-3 | |

Computer science majors must earn a C or better in all CSC core courses.

*Mathematics* (7 hours)

| | |
|---|---|
| MAT 111 Elementary Statistics | 3-0-3 |
| MAT 201 Calculus I | 4-0-4 |

***Elective Courses***

*Advanced Study* (12 hours)

An additional 12 hours of computer science course work chosen from among the following:

| | |
|---|---|
| CSC/MAT 319 Combinatorial Mathematics | 3-0-3 |
| CSC 333 Imbedded & Real-Time Microprocessor Interfacing & Control | 2-2-3 |
| CSC 340 Operating Systems | 3-0-3 |
| CSC/IS 361 Systems Analysis and Design **or** CSCIS 362 Database Management Systems | 3-0-3 |
| CSC 404WI Org. of Programming Languages | 3-0-3 |
| CSC 450 Net-centric Computing **or** CSC 461/ IS 461Data Communications & Networking | 3-0-3 |

*Mathematics* (3-4 hours)

One additional mathematics course chosen from among the following:

| | |
|---|---|
| MAT 202 Calculus II | 4-0-4 |
| MAT 303 Linear Algebra | 3-0-3 |
| CSC/MAT 319 Combinatorial Mathematics | 3-0-3 |

***Practical work experience***

Students must engage and document a practical work experience. Common ways to satisfy this requirement include academic internship (3 semester hours) and cooperative (co-op) work experience (80 work hours). Alternate methods may be proposed by the student and submitted to the department for approval.

**Appendix B. Information Systems Major Requirements**

***Note:*** The credit hours for each course is listed as *X-Y-Z* where *X* denotes the lecture contact hours per week, *Y* denotes the laboratory contact hours per week , and *Z* denotes the total student credit hours for the course.

***Business Precore*** (25 hours)

| | |
|---|---|
| ACC 201 Principles of Financial Accounting | 3-0-3 |
| ACC 202 Principles of Managerial Accounting | 3-0-3 |
| BUS 107 Business Information Management | 3-0-3 |
| BUS 211 Business Statistics | 3-0-3 |
| ECO 110 Principles of Economics I | 3-0-3 |
| ECO 210 Principles of Economics II | 3-0-3 |
| MAT 201 Calculus I | 4-0-4 |
| MAT 210 or 311 Statistics | 3-0-3 |

***Business Core*** (21hours)

| | |
|---|---|
| BUS 210 Legal Environment of Business | 3-0-3 |
| BUS 316 Information Systems | 3-0-3 |
| BUS 320 Principles of Marketing | 3-0-3 |
| BUS 330 Principles of Finance | 3-0-3 |
| BUS 350 Principles of Management | 3-0-3 |
| BUS 453 Operations Management | 3-0-3 |
| BUS 499 Global Business Strategy | 3-0-3 |

***Information Systems Concentration*** (24 hours)

*Computer Science Requirements* (12 hours)

| | |
|---|---|
| CSC 140 Introduction to Programming | 3-0-3 |
| CSC 310 Mathematical Programming in C++ | 3-0-3 |
| CSC/MAT 317 Discrete Mathematics | 3-0-3 |
| and **one** of the following | 3-0-3 |
| CSC 230 Advanced Programming Techniques | |
| CSC 401 Assembler Language | |
| CSC 404WI Org. of Programming Languages | |
| CSC 410 Data Structures I | |

**Note:** All of these courses except CSC 404 are being phased out of the CSC curriculum.

*Information Systems Requirements* (12 hours)

| | |
|---|---|
| BUS 361 Systems Analysis and Design | 3-0-3 |
| BUS 362 Database Management Systems | 3-0-3 |
| BUS 461 Data Communications and Networking | 3-0-3 |
| BUS 469 Systems Development Project | 3-0-3 |

**Appendix C. Decision Science Major Requirements**

***Note:*** The credit hours for each course is listed as *X-Y-Z* where *X* denotes the lecture contact hours per week, *Y* denotes the laboratory contact hours per week , and *Z* denotes the total student credit hours for the course.

***Business/Economics*** (15 hours)

| | |
|---|---|
| ACC 201 Principles of Financial Accounting | 3-0-3 |
| BUS 320 Principles of Marketing | 3-0-3 |
| BUS 330 Principles of Finance **or** BUS 334 Fundamentals of Investments | 3-0-3 |
| BUS 35OW1 Principles of Management | 3-0-3 |
| ECO I 10 Principles of Economics I | 3-0-3 |

***Mathematics*** (17 hours)

| | |
|---|---|
| MAT 201 Calculus I | 4-0-4 |
| MAT 202 Calculus II | 4-0-4 |
| MAT 203 Multivariable Calculus | 3-0-3 |
| MAT 304 Differential Equations | 3-0-3 |
| MAT 311 Probability & Statistics | 3-0-3 |

***Computer Science*** (15 hours)

| | |
|---|---|
| CSC 120 Principles of Computer Science I | 3-2-4 |
| CSC 121 Principles of Computer Science II | 3-2-4 |
| CSC 219 Discrete Structures | 3-0-3 |
| CSC 220 Data Structures & Multilevel Machines | 3-2-4 |

***Area of Concentration*** (15 hours)

The student will complete 15 additional hours in the three foundational disciplines (business, mathematics, and computer science), with these courses chosen to develop expertise in a particular aspect of decision science. The specific courses chosen should be planned in consultation with a program advisor, and must either 1) be entirely in one of the three disciplines, or 2) be approved by the program advisor in each discipline. Students pursuing option 1) in the area of computer science are further required to take CSC 320 and CSC 490WI.

**Appendix D. Proposed Information Systems Major Requirements**

***General Education Courses*** (53 hours)

Include ECO 110 and MAT 145 here

***Business Precore*** (15 hours)

| | |
|---|---|
| ACC 201 Principles of Financial Accounting | 3-0-3 |
| ACC 202 Principles of Managerial Accounting | 3-0-3 |
| BUS 211 Business Statistics | 3-0-3 |
| ECO 210 Principles of Economics II | 3-0-3 |
| MAT 210 or 311 Statistics | 3-0-3 |

***Business Core*** (18 hours)

| | |
|---|---|
| BUS 210 Legal Environment of Business | 3-0-3 |
| BUS 320 Principles of Marketing | 3-0-3 |
| BUS 330 Principles of Finance | 3-0-3 |
| BUS 350 Principles of Management | 3-0-3 |
| BUS 453 Operations Management | 3-0-3 |
| BUS 499 Global Business Strategy | 3-0-3 |

***Information Systems Concentration*** (33 hours)

*IS Foundation Requirements* (18 hours)

| | |
|---|---|
| CSC 120 Principles of Computer Science I | 3-2-4 |
| CSC 121 Principles of Computer Science II | 3-2-4 |
| CSC 219 Discrete Structures | 3-0-3 |
| IS 230 Introduction to IS Technologies | 3-2-4 |
| IS 231 Information Systems Theory and Practice | 3-0-3 |

**Notes:** IS 230 fulfills the BUS 107 requirement in the business pre-core. IS 231 fulfills the BUS 316 requirement in the business core.

*Information Systems Advanced Requirements* (15 hours)

Choose five from:

| | |
|---|---|
| IS 361 Systems Analysis and Design | 3-0-3 |
| IS 362 Database Management Systems | 3-0-3 |
| IS 461 Data Communications and Networking | 3-0-3 |
| IS 460 IS Topics | 3-0-3 |
| IS 462 IS Implementations | 3-0-3 |
| IS 469 Systems Development Project | 3-0-3 |

**Appendix E. Descriptions of Shared Courses**

**Computer Science:**

**CSC 120 Principles of Computer Science I 3-2-4**

| |
|---|
| *Introduction to algorithms and computational problem solving:* algorithms, structured decomposition, top-down design |
| *Fundamental programming constructs:* simple data types, variables, expressions, assignment, simple I/O, control structures, functions, and parameter passing, blocks and scope |
| *Introduction to computer architecture:* machine language; fundamental computer organization and the fetch-execute cycle |
| *Representation of data:* binary, integers, characters, floating point, audio, graphics |
| *Elementary language translation:* compilers and interpreters, object code, executable code, loading and linking |
| *Introduction to software development methodologies:* structured design, testing and debugging, programming environments |
| *Introduction to operating systems:* Role and purpose, comparative operating system packages, resource management. |
| *Introduction to networks:* Role and purpose. Network services, client-server computing, and protocols. |
| *Social issues:* History and social impact of computing. |

**CSC 121 Principles of Computer Science II 3-2-4**

| |
|---|
| *Introduction to structured data:* records, file I/O, strings, arrays (includes fundamental sorting and searching, intro. to recursion) |
| *Overview of database:* motivation, use of query languages |
| *Comparative programming language paradigms:* imperative, object-oriented, functional, logical |

| | |
|---|---|
| *Object-oriented programming:* design, encapsulation and information hiding, separation of declaration and definition, classes, sub- and super- classes, inheritance, polymorphism |
| *Introduction to analytical and empirical analysis:* proofs of correctness, timing functions, asymptotic notations, best, average, and worst case behaviors; experimental timing behavior |
| *Fundamentals of event driven programming and graphical user interfaces* |
| *Introduction to software engineering:* software design, testing and maintenance, code re-use and libraries, group work skills |
| *Social issues:* Risks and liabilities; copyrights, intellectual property, and software piracy. |

**Information Systems:**

**IS 220 Introduction to IS Technologies 3-2-4**
Adopted from **IS97.2**

CATALOG Students will extend ability to be efficient and effective in knowledge work by applying information technologies to problem situations and by design and use of small information systems for individuals and groups.

SCOPE This course enables students to improve their skills as knowledge workers through effective and efficient use of packaged software. It covers both individual and group work. The emphasis is on productivity concepts and how to achieve them through functions and features in computer software. Design and development of solutions focus on small systems.

TOPICS End user systems versus organization systems; analysis of knowledge work and its requirements; knowledge work productivity concepts; software functionality to support personal and group productivity; organization and management of software and data; accessing organization data, accessing external data; selecting a computer solution; developing a macro program by doing; designing and implementing a user interface; developing a solution using database software; refining and extending individual and group information management activities.

EXPLANATION AND EXPECTATIONS
Students who have prerequisite end-user knowledge work skills will have an opportunity to extend their basic problem solving skills by undertaking, completing and using a sequence of more extensive "personal systems." The course has both a theoretical problem solving component and an equivalent component of structured supervised laboratory experience. The knowledge work tool set as well as local and wide area network telecommunications are the context for the problem domain.

**IS 221 Information Systems Theory and Practice 3-0-3**
Adopted from **IS97.3**

CATALOG Students who have a solid computer science foundation and have constructed personal information systems will be exposed to the theory of the IS discipline. Application of these theories to the success of organizations and to the roles of management, users and IS professionals are presented. Emphasizes the fit of technologies to organizational needs.

SCOPE This course provides an understanding of organizational systems, planning, development, and decision process including how information is used for decision support in organizations. It covers quality and decision theory, information theory, and practice essential for providing viable information to the organization. It outlines the concepts of IS for competitive advantage, data as a resource, IS and IT planning and implementation, quality management and reengineering, project management and development of systems, and end user computing.

TOPICS Systems theory and concepts; information systems and organizational systems; decision theory and how it is implemented by IT; quality management and reengineering; level of systems: strategic, tactical and operational; system components and relationships; information system strategies; roles of information and information technology; roles of people using, developing and managing systems; IS planning; human-computer interface; network and telecommunications systems, applications, and management; electronic commerce; implementation and evaluation of system performance; societal and ethical issues related to information systems design and use and the IS profession.

EXPLANATION AND EXPECTATIONS
Students who have end-user skills who have implemented personal productivity systems using knowledge work tools will be prepared to use the information systems theory presented in this course. The course presents the basic concepts for use in subsequent courses; the systems point of view, the organization and development of a system, information flows, the nature of information systems, and basic techniques for representing systems structure. Learning, goal setting and achieving, decision making and other characteristics of individuals, groups and teams are explored. Organizational models and planning are presented. Quality concepts are explained. IS planning and development activities are explored in the organizational context of management and users. Cross-functional management and user teams are discussed.