Integration of Mathematical Concepts in the Computer Science, Information Technology and Management Information Science Curriculum

Donald Heier, Kathryn Lemm, Mary Reed, Erik Sand
Department of Computer Science and Technology
University of Jamestown
Jamestown, ND 58405
dheier@uj.edu

Abstract

The University of Jamestown is a small liberal arts educational institution with a Department of Computer Science and Technology (CS&T) that offers Bachelor of Arts degrees in Computer Science, Information Technology, and Management Information Science. The curriculum for each of these degrees currently requires math courses which are taught by the Department of Mathematics using a general approach. This paper outlines a plan to replace these math course requirements with courses taught by the CS&T professors that integrate mathematical concepts with programming and computer technology concepts. It is expected that the creation of the courses would make mathematical concepts more relevant and valuable to students and better prepare students to apply those concepts as they progress through their degree programs and subsequently into their careers.

1 Introduction

The University of Jamestown is a small liberal arts educational institution with a Department of Computer Science and Technology (CS&T) that offers Bachelor of Arts degrees in Computer Science (CS), Information Technology (IT), and Management Information Science (MIS) (1). The curriculum for each of these degrees currently requires math courses which are taught by the Department of Mathematics using a general approach. These math courses are expected to provide the mathematical foundations for computer coursework. Each degree program has a set of math requirements that consists of courses in college algebra, statistics, trigonometry and/or calculus. CS&T professors have noticed that most students struggle to apply math concepts in the computer science courses as these math courses do not effectively support student learning in the computer and technology field. This paper outlines a plan to better integrate mathematics concepts in the CS, IT, and MIS programs in response to updated Association for Computing Machinery (ACM) standards and current research literature. The plan includes the development of two new courses and updates to the requirements for the CS major and minor, IT major and minor, and the MIS major.

2 Background Research

The primary motivation for these changes is based on updated ACM standards and research referenced by The Liberal Arts Computer Science (LACS) Consortium.

The ACM, in the decades since the 1960s, along with leading professional and scientific computing societies, has endeavored to tailor curriculum recommendations to the rapidly changing landscape of computer technology (2). As the computing field continues to evolve, and new computing-related disciplines emerge, existing curricula are updated. At the University of Jamestown, ACM guidelines have been used to keep our programs in computer science and information technology current and relevant. The most recent ACM guidelines for computer science were approved in December of 2013. We have also researched larger schools such as Stanford University and the efforts they have taken to keep their curriculum in line with what is happening in the field (3).

The ACM guidelines suggest that there is a ‘deep and beautiful connection between mathematics and many areas of computer science” (2). The ACM notes the full set of mathematics in computer science programs varies broadly by institution and that, for example, restrictions on the number of courses that may be included in a program may lead to mathematics requirements that are “specially circumscribed to CS majors.” The ACM goes on to suggest “Students moving on to advanced coursework in specific areas of computing will likely need focused mathematical coursework relevant to those areas.” They further state that “Such coursework requirements are best left to the discretion of the individual programs and the areas of CS they choose to emphasize.”

The 2013 Guidelines list five example CS programs and dozens of example courses (2). One of the example programs was from Stanford University. That program contained two example courses that we felt would be appropriate for our programs. Those courses were Mathematical Foundations of Computer Science and Probability Theory for Computer Scientists. The foundations course had much in common with the foundations course we currently offer, but was positioned as an introductory course in the CS program. Further research revealed that several CS programs had introductory coursework that extended beyond the introduction to programming course. This led us to design CS 130, Introduction to Computer Principles, as a course to deliver basic computer mathematics concepts along with an introduction to the computing topics that students will be studying as they progress in their programs.

The second course adopted from Stanford was “Probability Theory for Computer Scientists.” As Stanford developed this course, they determined that students found the content more relevant and valuable than a generalized course of probability theory (4). They believe the course “helps effectively prepare students for applying probability in computing contexts and using it as a tool for data analysis and modeling.” Several other Universities have used this Stanford course as a model for their own courses. The Computer Science department at Villanova University believed that adopting a similar course would “allow students to appreciate the role of probability and statistics in computing” (5). This led us to design CS230, Introduction to Algorithm and Data Analysis.

The LACS Consortium is a group of computer scientists who work to model, advance, and sustain the study of computer science in liberal arts colleges throughout the nation (6). The group is actively engaged in curriculum development, scholarly research, and other projects that seek to advance high-quality undergraduate computer science education within the special setting of liberal arts institutions. In December of 2013, a few of the LACS members published a paper titled “The Roles of Mathematics in Computer Science”. This paper suggests there is a gap between math’s practical and intellectual roles in computer science and that required mathematics courses align poorly with the needs of computer science (7). The authors propose that the “content of required mathematics is of limited relevance to computer science as a whole. The remaining balance is under-utilized in the computer science curriculum.” They state that “such a curriculum produces graduates who are ill-equipped to use mathematics in their professional careers.” The authors note three solutions to the problem: a) computer science educators can be alerted to the problem, b) mathematics requirements can be used more efficiently, c) mathematics can be integrated into computer science courses. The conclusion of the paper charges faculty to reform the role of mathematics in their curricula. They encourage departments to prune mathematics courses with limited application to computer science and ensure more computer science courses use mathematics to illuminate computing concepts. The CS&T department at the University of Jamestown is very interested in improving the math skills of its students and has decided the best approach would be to integrate mathematics and computer science concepts into courses offered and taught by the CS&T department.

There is another consideration that needs mentioning. Currently, students in our programs complete their math requirements at any point during their college career. For example, a computer science student may take statistics as a freshman, or during his or her last semester, as a senior. This eliminates the possibility of utilizing the statistics concepts learned from the math course in computer science coursework and providing examples and illustrations of how these concepts apply to computer science. Since the courses are offered by another department not related to the CS&T department, requirements and course sequencing are not easy to manage. At times, students are taking their math classes online from other institutions which also meant the content was not consistent for all of our students. This proposal is intended to solve this problem by allowing the computer science instructors to build up relevant mathematics skills in our students as they progress through the programs.

For several years, the CS&T faculty have felt that the way mathematics is integrated into our majors is ineffective and have been looking for a way to improve the application of mathematics concepts within our teaching lessons. The CS&T faculty expect the findings from the Sanford implementation that indicate that “students find the contextualized content of this class more relevant and valuable than general presentations of probability theory” will be replicated in our implementation of similar coursework.

3 Curriculum Modifications

The proposal includes the development of two new courses and updates to the requirements for the CS major and minor, IT major and minor, and the MIS major.

3.1 New Courses

Two courses will be added:

CS 130: Introduction to Computer Principles

This course will present an overview of several important areas of the computer field (e.g. binary, octal and hexadecimal numbers, Boolean algebra and logic, data representation and storage, networking, operating systems, machine architecture, functions and algorithms) while seeking to develop a practical, realistic understanding of the field and to prepare students for future courses. Ethical and legal aspects of areas such as Internet security, software engineering, and database technology are also introduced in order to enable students to be responsible users of technology. 3 Credits

CS 230: Introduction to Algorithm and Data Analysis

This course examines the application of probability in the computer science field and its use in the analysis of algorithms. Students will learn how probability theory has become a powerful computing tool and what current trends are causing the need for probabilistic analysis. Computer science examples are used throughout, in areas such as: computer networks, data and text mining, simulation, computer security, remote sensing, computer performance evaluation, software engineering, robotics, and data management. 3 Credits

3.2 Modifications to Computer Science Program

The current program description listing nine credits of math courses will be removed, as the two courses detailed above will be added to the requirements for the CS major (1). Students will still need to complete a math course as part of the general education requirements. CS 130 will also be added to the CS minor. Table 1 outlines the program requirements for the CS major.

Course No.	Description	Credits
CS 130	Introduction to Computer Principles	3
CS 170	Structured Programming	3
CS 180	Object-oriented Programming	3
CS 210	Computer Organization and Architecture I	3
CS 230	Introduction to Algorithm and Data Analysis	3
CS 300	Data Structures and Algorithms	3
CS 310	Computer Organization and Architecture II	3
CS 325	Networking	3
CS 330	Computer Foundations	3
CS 342	Database Development	3
CS 343	Database Management	3
CS 360	Comparison of Programming Languages	3
CS 365	Information Systems Security	3
CS 410	Operating Systems Design	3
CS 480	Senior Seminar	1
CS 481	Computer Science as a Profession	1
CS xxx	Two Additional Programming Languages	6
CS 3xx/4xx	Upper division CS elective	3
	Total Semester Credits	**53**

Table 1: Requirements for Computer Science Major

3.3 Modifications to Information Technology Program

The current catalog lists three math courses of which the student can choose one to meet a 3 credit requirement (1). This selection list will be removed and CS 130 will be added to the major requirement. Students will still need to complete a math course as part of the general education requirements. In the IT minor CS 130 will be added, replacing one of the elective options. See Table 2 for a complete listing of the proposed program requirements for the IT major.

Course No.	Description	Credits
CS 130	Introduction to Computer Principles	3
CS 140 or CS 240	Integrated Software Packages Advanced Office Applications	3
CS 170	Structured Programming	3
CS 210	Computer Organization and Architecture I	3
CS 232 or CS 251	Presentation & Publishing Software Graphics Development	3
CS 241	Introduction to Information Systems	2
CS 272	Java Programming I	3
CS 322	Windows Server I	3
CS 325	Networking	3
CS 341	Web Development I	3
CS 342	Database Development	3
CS 343	Data Base Management	3
CS 351	Web Development II	3
CS 352	PC Assembly & Troubleshooting	2
CS 422	Windows Server II	3
CS 423	UNIX/LINUX	3
CS 480	Senior Seminar	1
CS 481	Computer Science as a Profession	1
CS xxx	One Additional Programming Course	3
CS 3xx /4xx	Upper Division CS Electives	6
	Total Semester Credits	**57**

Table 2: Requirements for Information Technology Major

3.4 Modifications to Management Information Science Program

The current catalog has a listing of three math courses of which the student can choose one to meet a 3 credit requirement, plus a 3 credit requirement for Math 111 – College Algebra (1). This listing will be removed and CS 130 and CS 230 will be added to the major requirements. Students will still need to complete a math course as part of the general education requirements. See Table 3 for a complete listing of the proposed MIS major program requirements.

Course No.	Description	Credits
CS 130	Introduction to Computer Principles	3
CS 140 or CS 240	Integrated Software Applications Advanced Office Applications	3
CS 170	Structured Programming	3

CS 230	Introduction to Algorithm and Data Analysis	3
CS 272	Java Programming I	3
CS 342	Database Development	3
CS 343	Database Management	3
CS 365	Information Systems Security	3
CS 420	Systems Analysis Methods	3
CS 430	Systems Design	3
CS 440	Management of Information Systems	3
CS 480	Senior Seminar	1
CS 481	Computer Science as a Profession	1
CS 3xx/4xx	Upper division CS electives	2
CS xxx	Additional Programming Course	3
Acct 201	Principles of Accounting I	3
Acct 202	Principles of Accounting II	3
Econ 201 or Econ 202	Principles of Economics I: Microeconomics Principles of Economics II: Macroeconomics	3
Choose Two: Busn 320 Busn 321 Busn 351	Marketing Business Management Financial Management	6
	Total Semester Credits	**55**

Table 3: Requirements for Management Information Science Major

4 Expected Outcomes

The proposed changes of our curriculum are expected to improve the overall intended outcomes of our programs in CS, IT and MIS. Specifically, the intended outcomes for adding CS 130 and CS 230 to the programs are:

- Students will gain an introductory overview of the variety of computer science study topics in their freshman year.

- Students will be able to formulate mathematical proofs using logic well before their senior year.

- Students will be able to apply mathematical tools such as induction and recursion well before their senior year.

- Students will be able to recall key definitions from set theory well before their senior year.

- Students will be able to distinguish between various computational models by the midpoint of their degree.

- Students will be able to think critically on the difficulties of key questions in foundations of computer science by the end of their freshman year.

- Students will gain an understanding of machine data representation and storage by the end of their freshman year.

- Students will be able to understand the combinatorial nature of problems by understanding the multiple possible outcomes from real problems and how various factors may affect an outcome by the midpoint of their programs.

- Students will gain a working knowledge of probability theory and how these theoretical fundamentals can be successfully applied in practice by the midpoint of their programs.

- Students will gain an appreciation for probabilistic statements and how understanding the difference between various statements requires an understanding of probabilistic analysis by the midpoint of their programs.

- Students will be exposed to a variety of applications where solutions to problems might otherwise be out of reach without the tools that probability can bring to bear by the midpoint of their programs.

- Students will receive an introduction to machine learning and how this has grown to impact many applications in computing in their freshman year.

- Students will understand how machine learning focuses on analyzing large quantities of data to build models that can then be harnessed in real problems, such as filtering email, improving web search, understanding computer system performance, and predicting financial markets by the midpoint of their programs.

Assessment of the degree to which these objectives have been met will be measured by performance on unit and final exams, student discussions, and course evaluations by the students. A final exit survey is given to all graduating seniors and these outcomes will be evaluated by students in this survey. We currently do some pre and post-testing in our classes for assessment purposes and we have some baseline data we can use to compare future results on these tests.

5 Conclusion

For a number of years the faculty members in the CS&T Department have been searching for ways to encourage students to take mathematics classes early in their education so application can be made to the programming and other skills being learned in our three majors. When the new ACM standards emphasized integrating math concepts into classes taught in the computer science departments, we saw an opportunity to revise our curriculum and offer classes that are directly related to courses in computing. Since these courses will be taught by our department, we can schedule them so they can be taken concurrently with relevant computing classes. Pending final approval of this proposal by the faculty senate and full faculty, these changes will be implemented in the 2015-2016 academic year and evaluated over the next several years.

References

[1] University of Jamestown. (2014). *2014-2015 Academic Catalog*. Retrieved from: http://issuu.com/jamestowncollege/docs/catalog_14-15/1

[2] Association for Computing Machinery (ACM), IEEE-Computer Society - Joint Task Force on Computing Curricula. (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Retrieved from: http://www.acm.org/education/CS2013-final-report.pdf

[3] Mehran Sahami, Alex Aiken, and Julie Zelenski. (2010). *Expanding the Frontiers of Computer Science: Designing a Curriculum to Reflect a Diverse Field*. Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE '10). ACM, New York, NY, USA, 47-51.

[4] Sahami, M. (2011). *A Course on Probability Theory for Computer Scientists*. Proceedings of the 42nd Technical Symposium on Computer Science Education (SIGCSE ‘11). ACM, New York, NY, USA, 263-268

[5] Joyce, D. (2011). *Probability and Statistics for Computer Science – Who Should Teach It?* Proceedings of the 2011 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS2011)

[6]The Liberal Arts Computer Science (LACS) Consortium. (2014). *What We Do*. Retrieved from: http://lacs.myopensoftware.org/

[7] Douglas Baldwin, Henry M. Walker, and Peter B. Henderson. (2013). *The Roles of Mathematics in Computer Science.* ACM Inroads 4, 4 (December 2013), 74-80.