Towards Math Integration in the Computer Science and Technology Curriculum

Donald Heier, Kathryn Lemm, Mary Reed, Erik Sand
Department of Computer Science and Technology
University of Jamestown
Jamestown, ND 57405
dheier@uj.edu

Abstract

The University of Jamestown is a small liberal arts college 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. In the fall of 2014, we developed a curriculum modification plan and made significant changes to the mathematics requirements for these degree programs. The main objective of the changes was to replace the math course requirements with courses taught by the CS&T professors that integrate mathematical concepts with programming and computer technology concepts. Our expectation was that these changes 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. Now that two new courses are fully developed, this paper reports on the progress made in this transformation and provides specific details of the math intensive courses.

1 Introduction

The University of Jamestown is a small liberal arts college 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. In the fall of 2014, we developed a curriculum modification plan and made significant changes to the mathematics requirements for these degree programs. The main objective of the changes was to replace the math course requirements with courses taught by the CS&T professors that integrate mathematical concepts with programming and computer technology concepts. Our expectation was that these changes 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.

Now that two new courses are fully developed, the purpose of this paper is to report on the progress made in this transformation and to provide specific details of the new courses. The department has previously taught a junior level course in computer foundations (CS 330) which is described as a survey of discrete mathematical structures and its applications pertaining to the study of computer science and information technology. This course has not changed. We now have added two new courses which are CS 130: Introduction to Computer Principles and CS 230: Introduction to Algorithm and Data Analysis (1). The details of these new courses are provided in sections 3.1 and 3.2.

2 Background Research

The primary motivation for changes implemented in 2015 was 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 was very interested in improving the math skills of its students and 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 needed mentioning. Previously, 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 eliminated 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 was not easy to manage. At times, students were taking their math classes online from other institutions which also meant the content was not consistent for all of our students. This change intends to solve this problem by allowing the computer science instructors to build up relevant mathematics skills in our students as they progress through their majors.

For several years, the CS&T faculty have felt that the way mathematics was integrated into our majors was ineffective and were 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 Math Intensive Courses

The Computer Science and Technology department now teaches three classes that we consider math intensive and replace courses that would have traditionally been taught by a Mathematics department. These courses are CS130: Introduction to Computer Principles, CS 230: Algorithm and Data Analysis, and CS 330: Computer Foundations.

3.1 CS 130: Introduction to Computer Principles

The Introduction to Computer Principles course was designed to present a breadth-first overview of the computer science discipline. The core of this course is the study of algorithms. Students are introduced to computing hardware and software, networks, programming languages, applications, and mathematical concepts basic to computer science. At the completion of this course, students should be able to demonstrate familiarity with concepts such as algorithmic problem solving, abstraction, pseudo code, binary numbers, Boolean logic, the Von Neumann architecture, system software, use of computers in todays' society and the social, ethical, and legal issues raised by pervasive computer technology.

3.1.1 Textbook Selection

The textbook selected for this course is titled and 'Invitation to Computer Science' and is published by Cengage (8). The publisher describes the textbook as follows:

> This flexible, non-language-specific text provides a solid foundation using an algorithm-driven approach that's ideal for your students' first course in Computer Science. Expanded chapter exercises and practice problems, feature boxes and the latest material on emerging topics, such as privacy, drones, cloud computing, and net neutrality, keep your course in touch with current issues. Optional online language modules for C++, Java, Python, C#, and Ada correspond seamlessly with this edition and give you the flexibility of incorporating a programming language to expand concepts from the text. … The optional CourseMate provides helpful study tools, such as flashcards, quizzing, and games, as well as an online Lab Manual containing 20 laboratory projects that map directly to the main text.

3.1.2 Course Outline

The following is a list of the main topics covered in this course:

- An Introduction to Computer Science.
- Algorithm Discovery and Design.
- The Efficiency of Algorithms.
- The Building Blocks: Binary Numbers, Boolean Logic, and Gates.
- Computer Systems Organization.
- An Introduction to System Software and Virtual Machines.
- Computer Networks and Cloud Computing.
- Information Security.
- Intro to High Level Languages.
- Multiple Programming Paradigms.
- Artificial Intelligence.
- Computer Graphics and Entertainment
- Making Ethical Decisions

3.1.3 Instructor Comments

At the completion of the course, the instructor made the following notes in an effort to make improvements to the course and increase student learning for future groups.

- Students were unfamiliar with the binary nature of a computer system. This resulted in difficulty breaking down algorithms into step-by-step pseudocode statements. Start with some 'simpler' examples.

- Students were particularly interested in the unit on computer networks, cloud computing, and information security. Include current events and expand the cloud references to include information about how their data is secured on the college network. (Possible guest speaker from our IT department).

- An individual term paper and small group presentation was assigned in conjunction with the Ethics unit. Effort was made to instruct students on using tools in Word to assist with writing the term paper and also how to rehearse and organize a presentation.

3.2 CS 230: Algorithm and Data Analysis

The Algorithm and Data Analysis course was designed to introduce students to statistical concepts, so they can make educated decisions in computer and business careers. Additionally, students are taught how to interpret statistical results that computerized processes create, interpret and apply statistics in a business and managerial context, and utilize statistical software to assist business decision-making. Some of the topics covered in this course are numerical descriptive measures, basic probability, discrete probability distributions, sampling distributions, fundamentals of hypothesis testing, two-sample tests and one-way ANOVA, chi-Square tests and simple linear regression.

3.2.1 Textbook Selection

The textbook selected for this course is titled 'Business Statistics' and is published by Pearson (9). The publisher describes the textbook as follows:

> Statistics is essential for all business majors, and this text helps students see the role statistics will play in their own careers by providing examples drawn from all functional areas of business. Guided by principles set by major statistical and business science associations (ASA and DSI), plus the authors' diverse experiences, the Seventh Edition of Levine/Szabat/Stephan's Business Statistics: A First Course continues to innovate and improve the way this course is taught to all students. This brief version, created to fit the needs of a one-semester course, is part of the established Berenson/Levine series. MyStatLab provides users with countless opportunities to practice, plus statistics-specific resources and tools that enhance students' experience and comprehension.

3.2.2 Course Outline

The following is a list of the main topics covered in this course:

- Introduction to MyStatLab and Excel
- Defining and Collecting Data

- Organizing and Visualizing Variables
- Numerical Descriptive Measures
- Basic Probability
- Discrete Probability Distributions
- The Normal Distribution
- Sampling Distributions
- Confidence Interval Estimation
- Fundamentals of Hypothesis Testing: One-Sample Tests
- Two-Sample Tests and One-Way ANOVA
- Chi-Square Tests
- Simple Linear Regression

3.2.3 Instructor Comments

At the midway point of this course, the instructor has made the following notes in an effort to make improvements to the course and increase student learning for future groups.

- Students should have experience and comfort in using Excel as Excel is used to generate statistical values.
- Work in some small group projects and presentations earlier in the semester. An end-of-semester project is planned, but earlier projects would be beneficial.

3.3 CS 330: Computer Foundations

This course provides a survey of the mathematics that pertain to the study of computer science and information technology. At the completion of this course, the student should be able to demonstrate understanding of the following objectives developed from the Core-Tier I Objectives of the ACM Curriculum (2):

- Use examples to explain the basic terminology of functions, relations, and sets.
- Perform the operations associated with sets, functions, and relations
- Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context.
- Convert logical statements from informal language to propositional and predicate logic expressions.
- Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis, program correctness, database queries, and algorithms.
- Apply formal and/or informal logical reasoning to real problems, such as predicting the behavior of software or solving problems.
- Define and identify sequences using a recursive relationship
- Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions.

- Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application.
- Illustrate by example the basic terminology of graph theory, as well as some of the properties and special cases of each type of graph/tree.
- Demonstrate different traversal methods for trees and graphs, including pre-, post-, and in-order traversal of trees.
- Model a variety of real-world problems in computer science using appropriate forms of graphs and trees.
- Calculate probabilities of events and expectations of random variables for elementary problems such as games of chance.
- Differentiate between dependent and independent events.

3.3.1 Textbook Selection

The textbook selected for this course is titled and 'Mathematics for Information Technology and is published by Cengage Learning (10). The publisher describes the textbook as follows:

> This text delivers easy-to-understand and balanced mathematical instruction. Each chapter begins with an application, goes on to present the material with examples, and closes with a summary of the relevant concepts and practice exercises. With numerous illustrations included, you'll be able to understand the content from a number of different angles. Whether you're majoring in electronics, computer programming, or information technology, you'll find Mathematics for Information Technology to be a valuable resource

3.3.2 Course Outline

The following is a list of the main topics covered in this course:

- Sets.
- Logic and Proof.
- Binary and Other Number Systems.
- Straight Line Equations and Graphs.
- Solving Systems of Linear Equations Algebraically and with Matrices.
- Sequences and Series.
- Vectors.
- Probability.
- Statistics.
- Graph Theory.

3.3.3 Instructor Comments

This course has been in the computer science curriculum for many years. At University of Jamestown, it is taught in alternating years. It is scheduled to be taught in the spring of 2017. Some of the content that has been in this course, will now have been introduced in previous courses. This may affect the amount of time that is needed to cover the traditional topics and allow for additional material to be covered and applied in different ways.

4 Expected Outcomes and Student Perception

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 shall 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.
- Student 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 probability allows us to solve problems that 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 could use to compare future results on these tests.

We are currently teaching the second class in this series. Initial assessment data will not be finalized until the class concludes. However anecdotal evidence via student feedback is currently available. All first year students who took the initial class were successful in the course and have decided to remain in the program.

Students who are in the Algorithm and Data Analysis class have given feedback about the class. They appreciate that the topics are being taught in a way that relates to their chosen major. They like the practical application that is being utilized in teaching the class.

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. These changes were implemented during the 2015-2016 academic year and will continue to be evaluated over the next several years.

References

[1] University of Jamestown. (2016). *2015-2016 Academic Catalog.* Retrieved from: http://issuu.com/jamestowncollege/docs/catalog_15-16/1

[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. DOI=10.1145/1734263.1734279

[4] Sahami, M. (2011). *A course on probability theory for computer scientists.* Proceedings of SIGCSE 2011, the Forty-Second Technical Symposium on Computer Science Education (New York: ACM, 2011): 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. DOI=10.1145/2537753.2537777

[8] Schneider, G. M., & Gersting, J. (2015). *Invitation to Computer Science, 7th Edition*, Course Technology, ISBN: 978-1305075771.

[9] Berenson, M. L., Stephan, D. F., Levine, D. M., & Krehbiel, T. C. (2012). *Business Statistics: A First Course, 7th Edition*, Pearson Higher Ed.,ISBN: 978-0321979018

[10] Basta, M., Delong, S. Basta, N., (2013), *Mathematics for Information Technology, 1st Edition*, Delmar Cengage Learning, ISBN: 978-1111127831