ABET
Self-Study Report
for the
BACHELOR OF SCIENCE IN COMPUTER SCIENCE
at
Fitchburg State University
160 Pearl Street
Fitchburg MA 01420

June 10, 2019

CONFIDENTIAL

The information supplied in this Self-Study Report is for the confidential use of ABET and its authorized agents, and will not be disclosed without authorization of the institution concerned, except for summary data not identifiable to a specific institution.

TABLE OF CONTENTS

Program Self-Study Report for
CAC of ABET
Accreditation or Reaccreditation

BACKGROUND INFORMATION

A. Contact Information

Brady Chen, Department Chair
Computer Science Department
Fitchburg State University
160 Pearl Street
Fitchburg, Massachusetts 01420
xchen@fitchburgstate.edu
1-978-665-3434

Keith Williamson
Dean of Business and Technology
Fitchburg State University
160 Pearl Street
Fitchburg, Massachusetts 01420
kwilli51@fitchburgstate.edu
1-978-665-3731

B. Program History

The Fitchburg State Computer Science Department was founded in 1975. The Bachelor of Science in Computer Science (CS) degree was the original program offered by the department. A major in Computer Information Systems (CIS) was introduced in 2001. Both CS and CIS programs have been accredited since 10/1/2008. In 2014, the game programming concentration for CS major was created. The cybersecurity concentration was introduced in 2017.

C. Options

The computer science program also offers a concentration in game programming.

D. Program Delivery Modes

The primary delivery of instruction is by class lectures during the day. Some courses are also offered online. Four required courses, Introduction to Electronics, Digital Electronics, Computer Organization, and Microprocessors have dedicated laboratory facilities used in addition to lectures.

E. Program Locations

All core courses are taught on campus during the day. A small number of elective courses are also offered online (Cybersecurity Management, Introduction to Data Science, Data Exploration and Visualization).

F. Public Disclosure

The department web page (https://www.fitchburgstate.edu/academics/academic-departments/computer-science-dept/) contains our Program Education Objectives (PEOs) and Student Outcomes (SOs). Annual student enrollment and graduation data are accessible to the public through the Department of Institutional Research (https://www.fitchburgstate.edu/offices-services-directory/institutional-research-and- planning/institutional-research/).

G. Deficiencies, Weaknesses or Concerns from the Previous Evaluation and the Actions Taken to Address Them

Our last evaluation team identified two program weaknesses:

1. **Criterion 3 Student Outcomes.** The Student Outcomes Criterion requires the program enable students to attain, by the time of graduation, an understanding of professional, ethical, legal, security, and social issues and responsibilities. Though the curriculum addresses this characteristic, coverage of these topics is minimal. As a consequence, the program lacks the necessary strength of compliance with this criterion.

    a. This weakness was satisfied with the establishment of a required course entitled "Ethical Issues in Computer Science".
    b. The course, CSC 4102, was first offered during the Spring semester of 2015 and is offered every semester.
    c. Further details can be found in in our Report to the ABET Final Statement (see Appendix G).

2. **Criterion 6 Faculty.** This criterion requires that the faculty serving in the program be of sufficient number to maintain continuity, stability, oversight, student interaction, and advising. In addition to the regular teaching load of four undergraduate courses per semester, some faculty members teach as many as two additional graduate courses. Requiring faculty members to teach an excessive number of courses in a single semester may impact the ability of the faculty to maintain continuity, stability, oversight, student interaction, and advising.

    a. Faculty members teaching more than one graduate course in a semester are required to apply the additional courses to their day load.
    b. Dr. Ricky Sethi was hired as a full-time Assistant Professor during Fall, 2014. Dr. Sethi supports both the Computer Science and Computer Information Systems programs. An additional faculty position has been approved for Fall 2019. The new faculty member will support both the Computer Science and Computer Information Systems programs, and provide the means for the department to appoint a program coordinator for each the Computer Science and Computer Information Systems programs.
    c. Further details can be found in in our Report to the ABET Final Statement (see Appendix G).

GENERAL CRITERIA

CRITERION 1. STUDENTS

A. Student Admissions

Students can declare a major at admission before actually matriculating. Admitted students are automatically registered for courses based on the department recommendation. During summer academic advising sessions for admitted students, the department chair or a department faculty volunteer consults with incoming students, assists them, and makes necessary changes for the course registration.
Transfer students are advised on a continuing basis by the university advising center and the department if necessary.

A few freshmen are 'undeclared' and join our department later. Some students switch to Computer Science from other majors.

All are self-selected, as are most of those who choose to leave.

B. Evaluating Student Performance

Student performance is measured using a variety of instruments depending on the course. Programming assignments, exams, quizzes and formal presentations and reports are all part of the process used to determine student performance. Student progress is monitored through courses passed and grades earned in those courses. Although we assess many courses to measure how each course is meeting program goals, we do not attempt to measure whether each student passing a course successfully meets all of the goals we have set for the course.

A campus-wide database (called 'Web4') tracks each student's progress through the program requirements. In particular, a Web-based tool DegreeWorks in Web4 allow the students to build their educational plan and monitor their academic progress toward degree completion. The DegreeWorks is also used during meetings with their assigned department advisors each semester. Students are required to meet with their advisors before they can register for the following term. This is enforced with a 'registration pin', unique to each student and each semester, which they need to register on-line, or by the advisor's signature on a paper registration form. Web4 enforces prerequisites and will only allow students to register for a course if the prerequisite is met or in-progress. The instructor can waive a prerequisite when circumstances warrant it. Under special circumstances, only the department chair can waive program requirements.

C. Transfer Students and Transfer Courses

Fitchburg State has transfer agreements with institutions throughout the state of Massachusetts. A program called "MassTransfer" (www.mass.edu/masstransfer) is in place for students seeking to transfer academic credit among any of the Massachusetts community colleges, state universities and University of Massachusetts campuses. Through MassTransfer, a student who completed an associate degree is eligible to transfer up to 60 credits into a linked MassTransfer program at a Massachusetts state university or University of Massachusetts campus. Qualified students may also receive automatic satisfaction of most or all general education requirements, guaranteed transfer of credits, guaranteed admission, waiver of application fee, and a tuition discount. Students without an associate degree may still transfer their credits through "MassTransfer Block." Qualified students may transfer a block of courses (34 credits) which satisfy general education, core, or distribution requirements between institutions. However, completing the MassTransfer Block does not guarantee admission. The maximum number of transfer credits that will be accepted by the University for any degree program is 75.

Additional agreements with specific institutions enable transfer students to automatically receive credit for some of our core and elective courses. Information on these additional agreements is also available through the MassTransfer web site. For example, CIS majors may receive credit for Introduction to Programming and economics courses taken at Bunker Hill Community College if they have opted for the Computer Information Systems concentration, taken the appropriate courses, and graduated with the associate degree.
Similarly, CS majors may receive credit for calculus, physics, CS 1 and CS 2 taken at Bunker Hill Community College if they have opted for the Computer Science concentration, taken the appropriate courses, and graduated with the associate degree.

More recently, Mount Wachusett Community College established a transfer track in their CIS program based on our CS and CIS program requirements. Students transferring into our program from this track are assured that their earned credits are optimized.

Students changing majors or transferring from institutions with which we have no transfer agreement are likely to graduate from the University with more than the minimum 120 credits. This may happen because some of the courses they transfer in do not directly map to our core. When no transfer agreement exists course transfers are handled on a case-by-case basis. The department chair may review the transcript of a transfer student and assign credit for one or more of our courses based on the course syllabi and the grades earned by the student. In any circumstance, the student can still apply these courses as general (not CS- related) electives.

Students entering our program with AP credits in Computer Science may request skipping Computer Science 1.

D. Advising and Career Guidance

Each full-time faculty member is required to schedule three hours weekly to meet with students for both academic and course advising. During these meetings students can receive additional one-on-one instruction outside of the classroom.

Each student in the major is assigned an advisor from the department faculty. Each advisor has both Computer Science and Computer Information Systems majors as advisees. All of the faculty members are familiar with the graduation requirements for both majors. In addition, they use DegreeWorks in Web4 (a campus-wide academic database) for reports of graduation readiness. The University has instituted a new program called the Student Success Collaborative (SSC) that allows faculty to establish advising campaigns and track student progress. The department has prepared a 4-year plan of study and a checklist for each major to assist advisors in determining whether students are on track. Additional office hours are set aside for a three-week period before registration begins to assist the advisees with course selection for the coming semester. Continuing students are required to meet with their advisors during the advising period to plan classes for next and future semesters. They must obtain their "registration pin" from the advisor, which is needed for registering on-line. The registration pin is unique to each student, and it changes each semester.

Some career advising happens during these semi-yearly sessions, and some occurs in or after class. We bring alumni to campus to talk to students about the real world from time to time. Often the speaker will leave with several resumes. The university maintains a Career Services Center that presents workshops to help students prepare for employment.

First year students obtain class assignments from the advising center. The course assignments are based on departmental recommendations. Faculty are available to help students adjust their schedules during summer months.

E. Work in Lieu of Courses

Students entering Fitchburg State University with college-level training or experience can be tested on college course material to earn credit toward their degree. The Fitchburg State University examination program is especially valuable for individuals who have had learning experiences outside the college classroom (employment experience, life experience, independent study, etc.) which may come to bear upon their formal academic training. It is possible to gain up to 60 college credits through the program.

On the College-Level Examination Program (CLEP), Fitchburg State University adheres to the standards established by the American Council on Education granting credit for tests on which a score of 50 has been achieved. This credit is awarded only to students enrolled in degree programs at Fitchburg State University.

In addition, the University has a Life Experience Credit Program (LECAP) to earn credit for courses based on their life experience. A student who takes this option is assigned a faculty member to evaluate a portfolio prepared by the student describing the work. The student pays full tuition for the credit-hours, and the faculty member assigns a grade based on the submitted portfolio.

F. Graduation Requirements

Requirements for Bachelor of Science in Computer Science include the following courses:

CSC/ MATH 1900 Discrete Mathematics
CSC 1500 Computer Science I
CSC 1550 Computer Science II
CSC 1600 Introduction to Electronics
CSC 1650 Digital Electronics
CSC 2560 Systems Programming
CSC 2600 Computer Organization
CSC 3011 Data Modeling and Database Design
CSC 3100 Operating Systems
CSC 3200 Programming Languages
CSC 3600 Microprocessors
CSC 3700 Algorithms and Data Structures
CSC 4102 Ethical Issues in Computer Science
CSC 4400 Software Engineering
MATH 1300 Pre-Calculus (if needed)
MATH 1800 Business Statistics
MATH 2300 Calculus I
MATH 2400 Calculus II
MATH 2600 Linear Algebra

One additional Math course at or above 3000 level
PHYS 2300 General Physics I

PHYS 2400 General Physics II
SPCH 1000 Introduction to Speech Communication
Four additional CS electives at or above 3000 level

In addition, there are university-wide Liberal Arts and Science (LA&S) requirements which apply to all majors. Some of those requirements are automatically met by the CS requirements; for example, the computer literacy and mathematics requirements are met and exceeded by the CS degree requirements.

Students and their advisors can use the online Degreeworks software to obtain a report that lists which requirements have been met and which remain. The registrar uses the same system to check the transcripts of students who have filed a graduation form indicating their intention to graduate at the end of the semester.

The department provides a Four-year Study Plan (see Appendix E) for the program which ensures a timely completion of degree requirements (120 credits). However, some students graduate with more than 120 credits for a number of reasons such as: additional major/minor, switching majors, personal interest in other areas (including from within our department), and transfer credits that do not fit the overall curriculum.

G. Transcripts of Recent Graduates

Transcripts are chronological for each semester. The student's major and concentration are stated. Each transcript provides the total credits earned from Fitchburg State University, credits transferred, and GPA based on the credits earned from the university.

CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES

A. Mission Statement

Fitchburg State University is committed to excellence in teaching and learning and blends liberal arts and sciences and professional programs within a small college environment. Our comprehensive public university prepares students to lead, serve, and succeed by fostering lifelong learning and civic and global responsibility. A Fitchburg State education extends beyond our classrooms to include residential, professional, and co-curricular opportunities. As a community resource, we provide leadership and support for the economic, environmental, social, and cultural needs of North Central Massachusetts and the Commonwealth.

B. Program Educational Objectives

CSPEO-1	Apply what they have learned about computer hardware, programming languages and the fundamental mathematical and engineering principles to computer system functions and software development.
CSPEO-2	Apply what they have learned about the theoretical foundations of computing machinery to understand and use new technologies as they are introduced.
CSPEO-3	Apply the skills they have learned to achieve and maintain professional careers in computer science and across other disciplines.
CSPEO-4	Apply what they have learned to behave ethically, professionally and provide leadership in their professional careers.
CSPEO-5	Apply communication skills including oral and written presentations.
CSPEO-6	Collaborate in group-projects.
CSPEO-7	Continue learning after graduation.

C. Consistency of the Program Educational Objectives with the Mission of the Institution

Program Educational Objectives	Mission Alignment Strength (0-2)
CSPEO-1 Apply what they have learned about computer hardware, programming languages and the fundamental mathematical and engineering principles to computer system functions and software development.	1
CSPEO-2 Apply what they have learned about the theoretical foundations of computing machinery to understand and use new technologies as they are introduced.	1
CSPEO-3 Apply the skills they have learned to achieve and maintain professional careers in computer science and across other disciplines.	1
CSPEO-4 Apply what they have learned to behave ethically, professionally and provide leadership in their professional careers.	2
CSPEO-5 Apply communication skills including oral and written presentations.	1
CSPEO-6 Collaborate in group-projects.	1
CSPEO-7 Continue learning after graduation.	2

D. Program Constituencies

Program constituencies include …

1. the students in the program, who hope to be prepared with the knowledge and skills for satisfying careers;
2. the students' potential customers and employers, both here in North-Central Massachusetts, and throughout the world, who expect effective workers and prospective leaders;
3. the students' families, who are often making a big investment;
4. the faculty, whose interactions with the students as they grow and learn can be extremely rewarding;
5. the administration and staff, whose primary goal is to assist the students and faculty in the logistics of running the institution.

	Alignment Strength (0-2)				
Program Educational Objectives	**Students**	**Employers**	**Family**	**Faculty**	**Staff**
CSPEO-1 Apply what they have learned about computer hardware, programming languages and the fundamental mathematical and engineering principles to computer system functions and software development.	2	2	1	2	1
CSPEO-2 Apply what they have learned about the theoretical foundations of computing machinery to understand and use new technologies as they are introduced.	2	2	1	2	1
CSPEO-3 Apply the skills they have learned to achieve and maintain professional careers in computer science and across other disciplines.	2	2	2	2	1
CSPEO-4 Apply what they have learned to behave ethically, professionally and provide leadership in their professional careers.	2	2	1	2	1
CSPEO-5 Apply communication skills including oral and written presentations.	2	2	1	2	1
CSPEO-6 Collaborate in group-projects.	2	2	1	2	1
CSPEO-7 Continue learning after graduation.	2	2	1	2	1

E. Process for Review of the Program Educational Objectives

The Program Educational Objectives are reviewed and revised in department faculty meetings. For example, we added a CS life-long learning objective during the October 2 and 9 meetings in Fall 2012. We recently updated CSPEO-3 to include the application of skills across other disciplines. Our discussion was inspired by our Industrial Advisory Board meeting in the Spring of 2018.

CRITERION 3. STUDENT OUTCOMES

A. Student Outcomes

The student outcomes CSSO-1 through CSSO-8 are listed in the table below. CSSO-1 through CSSO-6 are defined in Criterion 3 of ABET Criteria Version 2.0. CSSO-7 and CSSO-8 are additional outcomes for the computer science program.

	Program Student Outcomes
CSSO-1	Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
CSSO-2	Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
CSSO-3	Communicate effectively in a variety of professional contexts.
CSSO-4	Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
CSSO-5	Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
CSSO-6	Apply computer science theory and software development fundamentals to produce computing-based solutions. [CS]
CSSO-7	Demonstrate the ability to learn after leaving the university.
CSSO-8	Demonstrate the ability to design and implement digital logic circuits and apply this knowledge to the understanding of a computer's organization and architecture.

B. Relationship of Student Outcomes to Program Educational Objectives

The table below shows the relationship of Student Outcomes (CSSOs) to the Program Educational Objectives (PEO).

Program Student Outcomes	PEO Alignment Strength (0-2)						
	PEO	PEO	PEO	PEO	PEO	PEO	PEO
CSSO-1 - Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.	2	2	2	0	1	1	2
CSSO-2 - Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.	2	2	1	0	1	1	1
CSSO-3 - Communicate effectively in a variety of professional contexts.	1	1	2	2	2	2	1
CSSO-4 - Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	0	1	1	2	0	1	1
CSSO-5 - Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. [CS]	1	1	2	2	2	2	1
CSSO-6 - Apply computer science theory and software development fundamentals to produce computing-based solutions.	2	1	2	2	2	2	2
CSSO-7 - Demonstrate the ability to learn after leaving the university.	1	2	2	1	2	1	2
CSSO-8 - Demonstrate the ability to design and implement digital logic circuits and apply this knowledge to the understanding of a computer's organization and architecture.	2	2	1	0	1	1	1

C. Publication of Student Outcomes

The department documents our assessment intentions in a multi-sheet spreadsheet. We keep current and historical versions in a shared folder on the web, accessible to the entire department faculty, and essentially no one else. Program educational objective and stated student outcomes are publicly available from the department web site (https://www.fitchburgstate.edu/uploads/files/ComputerScienceDept/CS-Program- Educational-Objectives-2019.pdf). These are statements that describe what students are expected to know and/or be able to do by the time of graduation. If students have achieved these outcomes, it is anticipated that they will be able to achieve the educational objectives after graduation.

CRITERION 4. CONTINUOUS IMPROVEMENT

A. Student Outcomes

Twelve key courses were used for assessment purposes. Instructors for the 12 key courses gather assessment data every other year according to the schedule shown below. This schedule provides a complete program assessment every two years. Thus, since fall of 2015 we have completed two assessment cycles. For cycle 1 (Fall 2015 through Spring 2017) assessments were obtained for 9 of 12 key courses. A faculty member responsible for two courses retired and did not leave sufficient student data for assessment purposes (C2600 and C3600) during cycle 1. For cycle 2 (Fall 2017 through Spring 2019), assessment data was gathered for all 12 courses. Assessment tools align with course objectives and the number of objectives varies from 5 to 9 depending on the course. Student performance related to each objective is assessed by various tools embedded within each key course. The tools used to assess student learning of any given course objective may consist of quizzes (Q), exams (E), tests (T), homework (H), assignments (A), final exam questions (F), projects (P), lab exercises (L), final presentations (FP) or a combination of these. Student grades on each tool associated with each objective for each of the key courses are used to compute a score for each objective. A percentile rank of students (generally 70% to 80%) scoring above a particular threshold score (generally 70%) is used to identify areas requiring improvement. It is important to note that our class sizes are often small (24 maximum; many classes have enrollments less than 18). Smaller classes may have difficulty meeting an 80 percentile criteria for every course objective.

Assessments occur over a two-year cycle. During this period 12 key courses contribute to the assessment. One of these courses is offered in the Mathematics Department. The key courses used for assessment are:

CSC 1600 Introduction to Electronics
CSC 1650 Digital Electronics
CSC 1900 Discrete Math
CSC 2560 Systems Programming
CSC 2600 Computer Organization
CSC 3100 Operating Systems
CSC 3200 Programming Languages
CSC 3600 Microprocessors
CSC 3700 Algorithms and Data Structures
CSC 4102 Ethical Issues in Computer Science
CSC 4400 Software Engineering
MATH 2600 Linear Algebra

These 12 courses cover all of our student outcomes. Most items are assessed in more than one course as shown in the table below.

Table mapping key courses to student outcomes.

CS Student Outcomes	C3100	C3200	C2560	C3700	C4400	C1600	C1650	C2600	C3600	C1900	C4102	M2600
CSSO-1	X	X	X	X						X		X
CSSO-2					X			X	X			
CSSO-3					X	X						
CSSO-4					X						X	
CSSO-5					X				X			
CSSO-6					X				X			
CSSO-7		X		X	X	X	X	X	X			
CSSO-8							X	X				

The schedule for course assessments (as implemented) is shown in the table below.

	2015		2016		2017		2018		2019	
CS Outcomes	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall
CSSO-1		M2600		C3100		M2600		C3100		
		C3200 C2560	C3700 C1900				C3700 C1900	C2560 C3200		
CSSO-2				C2600	C3600 C4400		C3600	C2600	C4400	
CSSO-3				C1600	C4400			C1600	C4400	
CSSO-4			C4102		C4400		C4102		C4400	
CSSO-5					C3600 C4400		C3600		C4400	
CSSO-6					C4400				C4400	
CSSO-7			C3700				C3700			
		C3200		C1600	C3600 C4400		C3600	C1600 C3200	C4400	
CSSO-8			C1650	C2600				C2600	C1650	

Assessment Data Tables (Fall, 2015 to Spring, 2019)

Cycle 1: Fall, 2015 through Spring, 2017

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Solve systems of linear equations	Fall1	M2600		80% should score >= 70		Was not assessed by mathematics professor.
Finite dimensional vector spaces	Fall1	M2600		80% should score >= 70		
Perform linear transformations	Fall1	M2600		80% should score >= 70		
Bases and linear independence	Fall1	M2600		80% should score >= 70		
Eigenvalues and eigenvectors	Fall1	M2600		80% should score >= 70		
Functional, logical & procedural paradigms	Fall1	C3200	A1, T1	80% should score >= 70	90%	
Programming paradigm strengths and weaknesses	Fall1	C3200	A3, T2, F	80% should score >= 70	96%	
Implementation structures for the paradigms	Fall1	C3200	A2, T2, F	80% should score >= 70	82%	
Functional programming using Scheme	Fall1	C3200	A2 T3	80% should score >= 70	89%	
Logic programming using Prolog	Fall1	C3200	A4, F	80% should score >= 70	95%	
Sorting Algorithms	Spring1	C3700	A1;T 1	80% should score >= 70	76%	One large algorithm assignment from this semester will be broken down into several smaller assignments with additional instructions in future semesters. An additional set of exercises on pointers will be provided to help students do better with dynamic data structures.
Graph Algorithms	Spring1	C3700	A2;T 2	80% should score >= 70	92%	
Cryptographic Algorithms	Spring1	C3700	A3;T 3	80% should score >= 70	100%	
Dynamic data structures	Spring1	C3700	A4;T 4	80% should score >= 70	76%	

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Complexity theory	Spring1	C3700	T5	80% should score >= 70	92%	
Design paradigms	Spring1	C3700	T5;F	80% should score >= 70	100%	
Digital representation of data	Spring1	C1650	x	70% should score >= 70		Data missing due to Prof. Taylor's retirement.
Boolean algebra	Spring1	C1650	x	70% should score >= 70		
Combinational circuit analysis	Spring1	C1650	x	70% should score >= 70		
Combinational circuit implementation	Spring1	C1650	x	70% should score >= 70		
Sequential circuit analysis	Spring1	C1650	x	70% should score >= 70		
Sequential logic design	Spring1	C1650	x	70% should score >= 70		
Program a CPLD	Spring1	C1650	x	70% should score >= 70		
Plan and implement a logical design	Spring1	C1650	x	70% should score >= 70		
Professional and ethical responsibilities	Spring2	C4400	A2	80% should score >= 70	90%	Enforce an additional prerequisite: Web programming with Java to give students additional experience with implementation of the term project.
Software Engineering processes and CASE tools.	Spring2	C4400	A4	80% should score >= 70	95%	
Requirements Analysis and documentation.	Spring2	C4400	P1,A1	80% should score >= 70	100%	
Architecture design and documentation.	Spring2	C4400	P3, P4	80% should score >= 70	75%	
Database design and documentation.	Spring2	C4400	P2,A3	80% should score >= 70	90%	

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
User interface design and documentation.	Spring2	C4400	P1,P 4	80% should score >= 70	80%	
Object oriented analysis, design and documentation.	Spring2	C4400	P3,P 4	80% should score >= 70	75%	
Implementing verification, validation and testing.	Spring2	C4400	P3,P 4	80% should score >= 70	75%	
Working in groups to complete a software project.	Spring2	C4400	P5,A 4,FP	80% should score >= 70	83%	
Voltage, current, power and energy	Fall2	C1600	F1-29	70% should score >= 70	100%	Assessment improved due to tutoring on calculator use and arithmetic in Lab 0. Poor performance on RC circuit questions due to final topic of the semester and no specific lab. Stress of completing the final project may also play a role. May apply to opamps as well. Establish RC circuit lab, make project optional, create practice problem set to complete.
DC circuit analysis	Fall2	C1600	F35-39,47 ,48	70% should score >= 70	73%	
Build and test electronic circuits	Fall2	C1600	Lab	70% should score >= 70	91%	
Electrical signals	Fall2	C1600	F30-34	70% should score >= 70	73%	
RC circuit analysis	Fall2	C1600	F40-43	70% should score >= 70	27%	
Operational Amplifier circuits	Fall2	C1600	F49-58	70% should score >= 70	55%	
Discrete semiconductor circuits	Fall2	C1600	F59-62	70% should score >= 70	82%	
Data representations / Digital logic design	Fall2	C2600	x	80% should score >= 70		Data missing due to Prof. Taylor's retirement.
Register transfer language (RTL)	Fall2	C2600	x	80% should score >= 70		
Hardwired controller design and implementation	Fall2	C2600	x	80% should score >= 70		

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Microprogrammed controller design	Fall2	C2600	x	80% should score >= 70		
Instruction set architecture, processing, assembly and pipelining	Fall2	C2600	x	80% should score >= 70		
Memory hierarchy, cache techniques and virtual memory	Fall2	C2600	x	80% should score >= 70		
I/O methods, interrupts, raid techniques, data compression	Fall2	C2600	x	80% should score >= 70		
RISC versus CISC machines	Fall2	C2600	x	80% should score >= 70		
Structured programming with C	Fall1	C2560	P1;T 1	80% should score >= 70	84%	
Dynamic arrays and linked lists	Fall1	C2560	P3;T 2;F	80% should score >= 70	91%	
Trees and pointer arithmetic	Fall1	C2560	P2;T 2;F	80% should score >= 70	82%	
Pass by value versus pass by reference	Fall1	C2560	P2;T 3;F	80% should score >= 70	93%	
File manipulation and IO methods	Fall1	C2560	P4	80% should score >= 70	78%	
Problem Analysis and Design	Fall1	C2560	P2-4;F	80% should score >= 70	90%	
UNIX systems and programming	Fall1	C2560	T4	80% should score >= 70	79%	
Boolean expressions and Truth tables	Spring1	C1900	T1;F	80% should score >= 70	89%	
Proof techniques	Spring1	C1900	T2;F	80% should score >= 70	86%	

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Boolean techniques in digital electronics	Spring1	C1900	T3;F	80% should score >= 70	86%	
Basic Set theory	Spring1	C1900	T4;F	80% should score >= 70	98%	
Basic Number theory	Spring1	C1900	T5;F	80% should score >= 70	90%	
Basic counting principles	Spring1	C1900	T6;F	80% should score >= 70	98%	
Graphs and trees	Spring1	C1900	T7;F	80% should score >= 70	99%	
Basic Computational theory	Spring1	C1900	T8;F	80% should score >= 70	100%	
Machine Architecture	Spring2	C3600	F + Labs	70% should score >= 70	100%	First instance of a modernized version of the class. Cost-effective ISA bus computers are no longer available. New labs use microcontrollers rather than PCs. There were only 5 students in the class. Assessments that don't meet the target are only one student short.
Assembly Language	Spring2	C3600	F + Labs	70% should score >= 70	100%	
CPU Hardware	Spring2	C3600	F + Labs	70% should score >= 70	60%	
Memory Interfacing	Spring2	C3600	F	70% should score >= 70	80%	
I/O Interfacing	Spring2	C3600	F + Labs	70% should score >= 70	60%	
Interrupts	Spring2	C3600	F + Labs	70% should score >= 70	80%	
DMA	Spring2	C3600	F	70% should score >= 70	80%	
Identify ethical issues in Computer Science	Spring1	C4102	A1, T1	80% should score >= 70	100%	
Recognize and evaluate ethical choices in a modern	Spring1	C4102	A2, T2	80% should score >= 70	100%	

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
computerized world.						
Professional and ethical responsibilities defined in the ACM Professional Code of Ethics	Spring1	C4102	A3, T3	80% should score >= 70	100%	
Intellectual property rights, privacy and civil liberties, cyber-security, social and ethical implications of new technologies.	Spring1	C4102	A4, T4	80% should score >= 70	100%	
Improve presentation skills	Spring1	C4102	A5, T5	80% should score >= 70	100%	
Principles and components of an Operating System	Fall2	C3100	H1;P 1;T;F	80% should score >= 70	73%	Use homework material in quizzes
Processes	Fall2	C3100	H2,3; P2;T; F	80% should score >= 70	73%	
CPU scheduling, deadlock detection and deadlock avoidance.	Fall2	C3100	H2,3; P2;T; F	80% should score >= 70	73%	
Memory management	Fall2	C3100	H4,5; F	80% should score >= 70	80%	
File systems	Fall2	C3100	H5;F	80% should score >= 70	93%	
Operating system security issues.	Fall2	C3100		80% should score >= 70		

Cycle 2 – Fall, 2017 through Spring, 2019

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Solve systems of linear equations	Fall1	M2600	Q1;T1; T2;F	70% should score >= 70	73%	
Finite dimensional vector spaces	Fall1	M2600	Q1;T1; F	70% should score >= 70	73%	
Perform linear transformations	Fall1	M2600	Q3;T1; F	70% should score >= 70	74%	
Bases and linear independence	Fall1	M2600	Q5;T2; F	70% should score >= 70	71%	
Eigenvalues and eigenvectors	Fall1	M2600	T2;F	70% should score >= 70	69%	
Vector spaces and subspaces	Fall1	M2600	Q6;T2; F	70% should score >= 70	64%	
Dimension and Coordinate systems	Fall1	M2600	T2;F	70% should score >= 70	69%	
Functional, logical & procedural paradigms	Fall2	C3200	A1, T1	80% should score >= 70	91%	
Programming paradigm strengths and weaknesses	Fall2	C3200	A3, T2, F	80% should score >= 70	95%	
Implementation structures for the paradigms	Fall2	C3200	A2, T2, F	80% should score >= 70	80%	
Functional programming using Scheme	Fall2	C3200	A2 T3 F	80% should score >= 70	92%	
Logic programming using Prolog	Fall2	C3200	A4, F	80% should score >= 70	89%	
Sorting Algorithms	Spring1	C3700	A1;T1	80% should score >= 70	76%	
Graph Algorithms	Spring1	C3700	A2;T2	80% should score >= 70	90%	

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Cryptographic Algorithms	Spring1	C3700	A3;T3	80% should score >= 70	99%	
Dynamic data structures	Spring1	C3700	A4;T4	80% should score >= 70	76%	
Complexity theory	Spring1	C3700	T5	80% should score >= 70	87%	
Design paradigms	Spring1	C3700	T5;F	80% should score >= 70	98%	
Digital representation of data	Spring2	C1650	F	70% should score >= 70	77%	Results improved over last cycle. Students complete HDL assignments in the lab but understanding is lacking when tested. Possibilities for improvement include adding a dedicated lecture, improved handout material, homework assignment, quiz to motivate.
Boolean algebra	Spring2	C1650	F	70% should score >= 70	95%	
Logic circuit analysis	Spring2	C1650	F	70% should score >= 70	82%	
Combinational circuit implementation	Spring2	C1650	F	70% should score >= 70	100%	
Combinational logic functions	Spring2	C1650	F	70% should score >= 70	95%	
Sequential logic design	Spring2	C1650	F / Ex3	70% should score >= 70	82%	
Programmable logic, HDL	Spring2	C1650	F / Lab	70% should score >= 70	55%	
Documentation and reporting	Spring2	C1650	Lab	70% should score >= 70	86%	
Professional and ethical responsibilities	Spring2	C4400	A2	80% should score >= 70	100%	Add prerequisite in data modeling (a new course) to give students the experience needed for the term project implementation.
Software Engineering processes and CASE tools.	Spring2	C4400	A3, P1	80% should score >= 70	83%	
Requirements Analysis and documentation.	Spring2	C4400	P1,P4	80% should score >= 70	97%	

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Architecture design and documentation.	Spring2	C4400	P3,P5	80% should score >= 70	80%	
Database design and documentation.	Spring2	C4400	P2	80% should score >= 70	100%	
User interface design and documentation.	Spring2	C4400	P1,P4	80% should score >= 70	97%	
Object oriented analysis, design and documentation.	Spring2	C4400	A3,P3, P5	80% should score >= 70	75%	
Implementing verification, validation and testing.	Spring2	C4400	A3,P3, P5	80% should score >= 70	75%	
Working in groups to complete a software project.	Spring2	C4400	P2,FP	80% should score >= 70	100%	
Fundamental electrical quantities	Fall2	C1600	F1-29	70% should score >= 70	93%	Calculator and arithmetic exercises have helped. RC circuits understanding has improved but that has resulted in decreased performance on switching circuits and op-amps. May be in part because the term project was dropped to reduce EoS stress. Try re-establishing the project next Fall. Exam portion of lab performance was not good. Try lab quizzes next Fall.
DC circuit analysis	Fall2	C1600	F35-39,47,48	70% should score >= 70	72%	
Build and test electronic circuits	Fall2	C1600	F + Lab	70% should score >= 70	71%	
Electrical signals	Fall2	C1600	F30-34	70% should score >= 70	85%	
RC circuit analysis	Fall2	C1600	F40-43	70% should score >= 70	70%	
Operational Amplifier circuits	Fall2	C1600	F49-58	70% should score >= 70	48%	
Switching circuits	Fall2	C1600	F	70% should score >= 70	67%	
Documentation & communication	Fall2	C1600	F59-62	70% should score >= 70	75%	
Digital representations/logic: transfer & computation	Fall1	C2600	F + P1	70% should score >= 70	92%	First time teaching after Prof. Taylor retired. Students did well on topics with hands-on

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
ALU realization	Fall1	C2600	P1	70% should score >= 70	100%	components. Topics without lab projects were marginal. Approaches to improvement include more frequent quizzing on memory, parallelism and caches.
Controller realization	Fall1	C2600	F + P1	70% should score >= 70	69%	
RTL / HDL / FPGA implementations	Fall1	C2600	F + P2-5	70% should score >= 70	85%	
ISA, logic for CPU operations	Fall1	C2600	F + P1	70% should score >= 70	77%	
Memory, parallelism, cache, etc.	Fall1	C2600	F	70% should score >= 70	54%	
RISC / CISC, Harvard / Princeton	Fall1	C2600	F	70% should score >= 70	69%	
Professional documentation/communication	Fall1	C2600	P 1-5	70% should score >= 70	92%	
Digital representations/logic: transfer & computation	Fall2	C2600		70% should score >= 70		Adjunct taught due to FT faculty shortage. No assessment performed. Fall1 assessment provided instead.
ALU realization	Fall2	C2600		70% should score >= 70		
Controller realization	Fall2	C2600		70% should score >= 70		
HDL / FPGA implementations	Fall2	C2600		70% should score >= 70		
ISA, logic for CPU operations	Fall2	C2600		70% should score >= 70		
Memory, parallelism, cache, etc.	Fall2	C2600		70% should score >= 70		
RISC / CISC, Harvard / Princeton	Fall2	C2600		70% should score >= 70		
Professional documentation/communication	Fall2	C2600		70% should score >= 70		

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Structured programming with C	Fall2	C2560	A1, T1	80% should score >= 70	84%	
Dynamic arrays and linked lists	Fall2	C2560	A3, T2, F	80% should score >= 70	93%	
Trees and pointer arithmetic	Fall2	C2560	A2, T2, F	80% should score >= 70	78%	
Pass by value versus pass by reference	Fall2	C2560	A2 T3 F	80% should score >= 70	76%	
File manipulation and IO methods	Fall2	C2560	A4	80% should score >= 70	89%	
Problem Analysis and Design	Fall2	C2560	A2, A3, A4 F	80% should score >= 70	87%	
UNIX systems and programming	Fall2	C2560	T4	80% should score >= 70	85%	
Boolean expressions and Truth tables	Spring1	C1900	T1;F	80% should score >= 70	91%	
Proof techniques	Spring1	C1900	T2;F	80% should score >= 70	87%	
Boolean techniques in digital electronics	Spring1	C1900	T3;F	80% should score >= 70	86%	
Basic Set theory	Spring1	C1900	T4;F	80% should score >= 70	93%	
Basic Number theory	Spring1	C1900	T5;F	80% should score >= 70	91%	
Basic counting principles	Spring1	C1900	T6;F	80% should score >= 70	100%	
Graphs and trees	Spring1	C1900	T7;F	80% should score >= 70	99%	
Basic Computational theory	Spring1	C1900	T8;F	80% should score >= 70	95%	
Machine Architecture	Spring1	C3600	F + Labs	80% should score >= 70	93%	With most of the class based on lab projects, topics that were not

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
Assembly Language	Spring1	C3600	F + Labs	80% should score >= 70	93%	demonstrated in the lab did not do well. Memory interfacing was covered but it was not a student focus for final exam. Use another instrument to accurately assess memory interfacing and a corresponding lab simulation exercise. Cover DMA in class followed by a quiz.
CPU Hardware	Spring1	C3600	F + Labs	80% should score >= 70	87%	
Memory Interfacing	Spring1	C3600	F	80% should score >= 70	47%	
I/O Interfacing	Spring1	C3600	F + Labs	80% should score >= 70	80%	
Interrupts	Spring1	C3600	F + Labs	80% should score >= 70	87%	
DMA	Spring1	C3600	-	80% should score >= 70		
Machine Architecture	Spring2	C3600		80% should score >= 70		Adjunct taught due to lack of full-time faculty. No assessment performed. See Spring 1 assessment instead.
Assembly Language	Spring2	C3600		80% should score >= 70		
CPU Hardware	Spring2	C3600		80% should score >= 70		
Memory Interfacing	Spring2	C3600		80% should score >= 70		
I/O Interfacing	Spring2	C3600		80% should score >= 70		
Interrupts	Spring2	C3600		80% should score >= 70		
DMA	Spring2	C3600		80% should score >= 70		
Identify ethical issues in Computer Science	Spring1	C4102	A1, T1	80% should score >= 70	100%	
Recognize and evaluate ethical choices in a modern	Spring1	C4102	A2, T2	80% should score >= 70	100%	

Course Objective	Term	Course	Tool	Target Percent	Actual Percent	Action to be taken
computerized world.						
Professional and ethical responsibilities defined in the ACM Professional Code of Ethics	Spring1	C4102	A3, T3	80% should score >= 70	100%	
Intellectual property rights, privacy and civil liberties, cyber-security, social and ethical implications of new technologies.	Spring1	C4102	A4, T4	80% should score >= 70	100%	
Improve presentation skills	Spring1	C4102	A5, T5	80% should score >= 70	100%	
Principles and components of an Operating System	Fall2	C3100	H1; P1;T1	70% should score >= 70	67%	New textbook and project changed to minix resulted in student issues.
Processes	Fall2	C3100	H2;P2, T1	70% should score >= 70	61%	
CPU scheduling, deadlock detection and deadlock avoidance.	Fall1	C3100	H2,3;P 2,3;T1, 2	70% should score >= 70	50%	
Memory management	Fall1	C3100	H4,5;T 2;F	70% should score >= 70	83%	
File systems	Fall1	C3100	F	70% should score >= 70	78%	
Operating system security issues.	Fall1	C3100		70% should score >= 70		

B. Continuous Improvement

Since our last interim report, we have continued our program-wide embedded assessment model. The assessment tools and targets are set by the individual instructors based on course objectives, course level, course complexity and previous assessment results.

Each course outline listed in the "Course Syllabi" section (Appendix A) contains a table showing how each course objective aligns with program student outcomes. The student outcomes are mapped to program educational objectives as given in the table under Criterion 3 section B.

For assessing courses in the CS program, we use a percentile above a threshold grade. This measure tells us the proportion of students meeting the threshold criteria and gives an indication of how well the student population performs with respect to each course objective (currently 80% or 70% depending on course).

During the first assessment cycle (Fall 2015 through Spring 2017), 75 course objectives were assessed. However, the data for 16 of these is unavailable. Of the remaining 59, 14 (24%) did not meet the target percentile which is improved over the second cycle of our previous report (39%). However, 12 of the 14 were within 10% of meeting assessment criteria. The remaining 2 objectives were associated with C1600 (Introduction to Electronics). Previously implemented corrective actions focused on encouraging numeracy early in the semester and adjusting assessment tools to compensate for long-term forgetfulness. Further actions were taken to reduce end-of-semester stress by making the final term project optional. Students are then able to focus more on the theoretical concepts (RC circuits and Operational amplifiers).

During the second assessment cycle (Fall 2017 through Spring 2019), 82 course objectives were assessed and 19 (23%) did not meet the target percentile. However, 14 of the 19 were within 10% of meeting criteria. The remaining 5 were distributed across 5 courses: C1600 (1/8), C1650 (1/8), C2600 (1/8), C3600 (1/6), and C3100 (1/5). In the previous cycle, C1600 had two major issues, one (RC Circuits, 27%) has been resolved in the current cycle (70%) and the other issue (operational amplifiers) did not markedly improve. New hands-on operational amplifier exercises will be implemented in the coming semester to help improve this score. As for C3100 (Operating Systems), the previous cycle demonstrated only minor issues. In the most recent cycle, the instructor decided to go with a new textbook and projects based on the Minix 3 operating system. This resulted in student assessment that did not meet the target for CPU scheduling and deadlock. More experience with these lab projects and the textbook should help resolve this issue in the future.

The student assessment of memory interfacing in C3600 (Microprocessors) did not meet the target. This result may be due to a combination of the assessment instrument (Final Exam) being administered at the end of the semester and no hands-on lab experiences associated with the topic. In the future, we will implement the assessment mid-semester, closer to the time the topic was covered in class. A similar issue was revealed in the assessment of C2600 (Computer Organization) where concepts related to memory organization and parallelism are not incorporated into lab exercises. Moving the assessment instrument closer to mid- semester may yield better results. In C1650, the assessment of understanding a hardware definition language (SystemVerilog) did not meet the assessment target. Students completed several lab exercises, however, they were unable to demonstrate understanding when tested. Enhanced in-class coverage and perhaps a quiz will be implemented in an attempt to improve the assessment of this topic.

Our assessment methods continue to be a work in progress. With each cycle, we learn more about how to fairly assess student learning and make changes to improve our program. The goal of course assessment is to take effective action to improve learning when student performance related to a course objective does not meet the criteria we establish.

The factors associated with student performance measured by any particular tool are numerous and can be viewed from multiple perspectives. In every course offering, faculty measure student performance throughout the semester using a variety of tools to determine the grade assigned to each student. Our embedded assessment methodology uses student performance on some of these tools to also attempt a determination of the percentage of students that demonstrate understanding/competence for each course objective. These tools, however, often serve a tertiary purpose: they provide student feedback throughout the semester to motivate further learning. This means that students may not actually gain understanding of a particular course objective until after they have failed an assignment used as an assessment tool.

We continue to work toward making our assessment tools a genuine reflection of student learning so we may have a fair assessment of our program. This work may include changing assessment tools, time of execution, weights given to individual tools or assessment criteria (threshold grade, target percentile or both). In addition, we may revise course objectives or the way content is delivered, and we may modify student exercises to emphasize particular course objectives.

There are many other aspects out of our control. In particular, student preparedness in mathematics seems to vary greatly. Although we would like to see every student choosing to major in computer science complete the program, many students move to other majors (often, to our CIS program) during the second year because they do not have the prerequisite mathematical interest/skill necessary for success. These students may be included in the assessment data for some courses.

In conclusion, we believe our assessment procedures and data demonstrate our commitment to maintaining the quality of our program. Data obtained during the first assessment cycle resulted in modifications to our teaching methods. The course flagged for action during the

first cycle showed some improvement during the second cycle. Course assessments for the second cycle flagged four additional courses. None of these classes had more than one objective flagged. Our review of this most recent data has resulted in a number of actions aimed at improving student learning in our program. Despite our best efforts to provide consistency in course delivery, we see variability in our assessment results between cycles. This variability may reflect differences between student cohorts. We will continue to adjust our teaching methods and/or assessment tools working toward consistency in our assessment data and improving student learning.

C. Additional Information

Assessment instrument examples will be available at the time of the site visit.

CRITERION 5. CURRICULUM

A. Program Curriculum

Table 5-1 Curriculum - BS Computer Science

Course	Required, Elective or a Selected Elective by an R, an E or an SE.[1]	Subject Area (Credit Hours)				Last Two Offerings	Section Enrollment: Student # (sections)
		Math & Sciences	Computing Topics **F**undamental **A**dvanced	General Education	Other		
Fall Freshman Year (16 credits)							
ENGL 1100 Writing 1	R			3		S19, F18	77(4), ?
CSC 1500 Computer Science I	R		3F			S19, F18	70(4), 80 (5)
MATH 1300 Precalculus (if needed)	R	3				S19, F18	102(5), ?
PHYS 2300 General Physics I (SMT)	R	4				S19, F18	67(5), ?
History LA&S elective (CTW)	SE			3			
Spring Freshman Year (16 credits)							
ENGL 1200 Writing II	R			3		S19, F18	626(26), ?
CSC 1550 Computer Science II	R		3F			S19, F18	65(4), 36(2)
CSC/MATH 1900 Discrete Math (SMT)	R	3				S19, F18	43(2), ?
PHYS 2400 General Physics II (SMT)	R	4				S19, F18	46(3), ?
EXSS 1000 LA&S elective (SMT)	R	3				S19, F18	311(12), ?
Fall Sophomore Year (17 credits)							
CSC 1600 Intro. To Electronics	R	4				F18, F17	36(2), 19(1)
CSC 2560 Systems Programming	R		3F			S19, F18	22(1), 37(2)
MATH 2300 Calculus I	R	4				S19, F18	48(2), ?
Literature LA&S elective (ARTS)	SE			3			
Literature LA&S elective (CTW)	SE			3			
Spring Sophomore Year (17 credits)							
CSC 1650 Digital Electronics	R		4F			S19, S18	24(1), 10(1)
CSC 3700 Algorithms & Data Structure	R		3A			S19, S18	22(1), 16(1)

Course	Required, Elective or a Selected Elective by an R, an E or an SE.[1]	Subject Area (Credit Hours)				Last Two Offerings	Section Enrollment: Student # (sections)
		Math & Sciences	Computing Topics **F**undamental **A**dvanced	General Education	Other		
MATH 2400 Calculus II	R	4				S19, F18	14(1), ?
SPCH 1000 Speech (ARTS)	R			3		S19, F18	90(4), ?
Behavior LA&S elective (CTW)	SE			3			
Fall Junior Year (16 credits)							
CSC 2600 Computer Organization	R		4F			F18, F17	12(1), 15(1)
CSC 3200 Programming Languages	R		3A			F18, F17	21(1), 21(1)
MATH 1800 Business Statistics	R	3				S19, F18	55(3), ?
CSC 3XXX CSC Elective	E		3A				
Free Elective					3		
Spring Junior Year (14 credits)							
CSC 3600 Microprocessors	R		4A			S19, S18	13(1), 14(1)
CSC 3100 Operating Systems	R		3A			F18, F17	20(1), 17(1)
MATH 2600 Linear Algebra	R	3				S19, F18	25(1), ?
CSC 3XXX CSC Elective	E		3A				
Free Elective					3		
Fall Senior Year (12 credits)							
CSC 3XXX CSC Elective	E		3A				
CSC 3XXX CSC Elective	E		3A				
Art/Music LA&S elective	SE			3			
Free Elective	SE				3		
Spring Senior Year (12 credits)							
CSC 4400 Software Engineering	R		3A			S19, S17	19(1), 10(1)
MATH 3XXX Math Elective	SE	3					
CSC 3XXX CSC Elective	E		3A				
CSC 4102 Ethical Issues in CS	R		1F			F18, F17	25(1), 8(1)

Course	Required, Elective or a Selected Elective by an R, an E or an SE.[1]	Subject Area (Credit Hours)				Last Two Offerings	Section Enrollment: Student # (sections)
		Math & Sciences	Computing Topics **F**undamental **A**dvanced	General Education	Other		
Add rows as needed to show all courses in the curriculum.							
TOTALS-ABET BASIC-LEVEL REQUIREMENTS		38	18F+31A =49	24	9		
OVERALL TOTAL CREDIT HOURS FOR COMPLETION OF PROGRAM	120						

1. **Required** courses are required of all students in the program, **elective** courses (often referred to as open or free electives) are optional for students, and
   selected elective courses are those for which students must take one or more courses from a specified group.
2. If math and science courses are chosen from a list indicate this and include information elsewhere on the courses that students may choose from.
3. For courses that include multiple elements (lecture, laboratory, recitation, etc.), indicate the maximum enrollment in each element. For selected elective courses, indicate the maximum enrollment for each option.

Instructional materials and student work verifying compliance with ABET criteria for the categories indicated above will be required during the campus visit.

The program's requirements are specified in Table 5-1 (above) indicating all the required courses for the program. Each course outline in the "Course Syllabi Section" (Appendix A) contains a table that shows how the course objectives help students attain the stated program outcomes. In section B of Criterion 3, we present a table that maps student outcomes to program educational objectives.

The syllabus of each required course in Appendix A has a table that maps each course objective to the program student outcomes. The table below shows how each key course contributes to each student outcome. Most student outcomes are covered by more than one key required course.

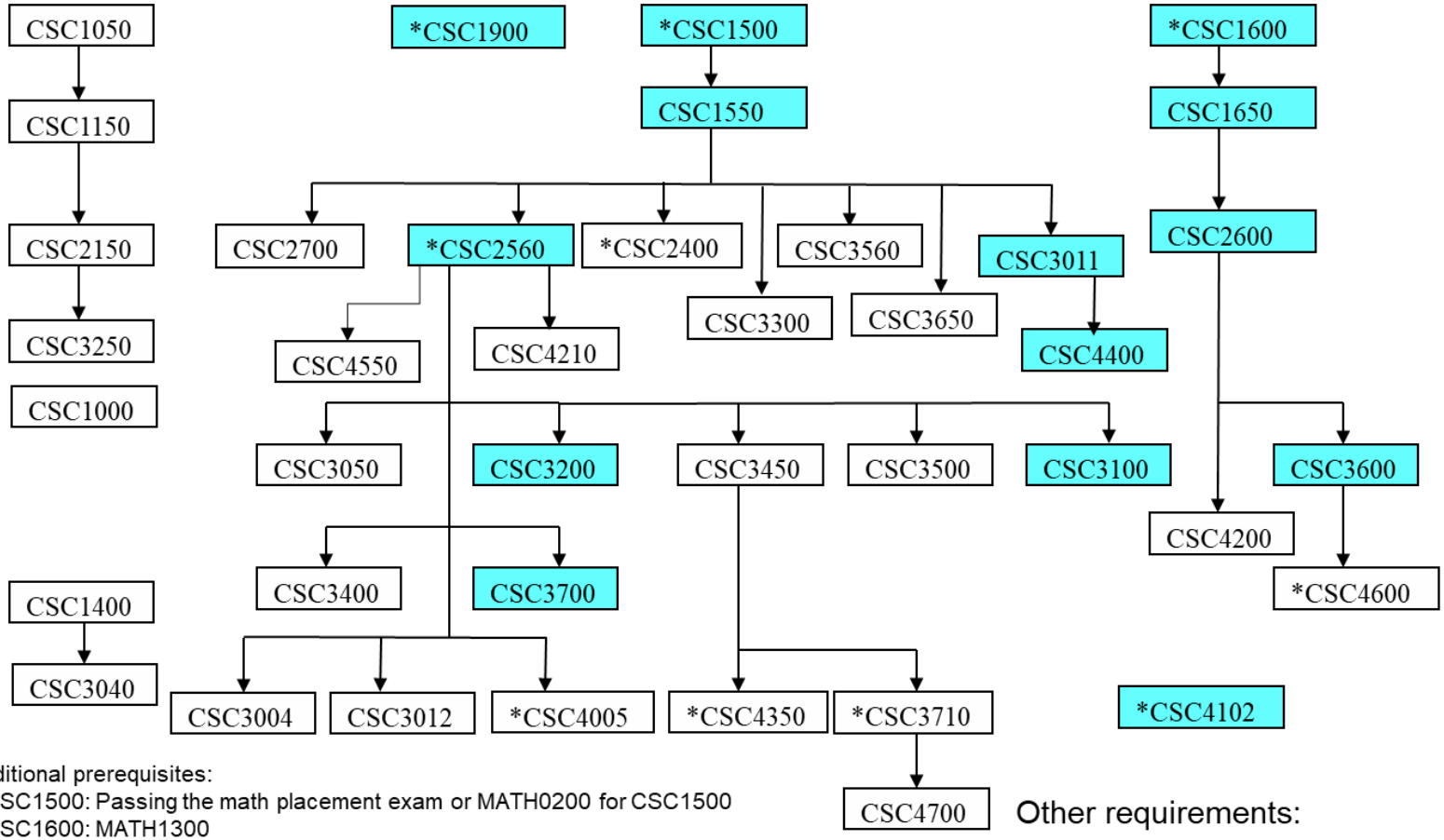
Table mapping key courses to program student outcomes.

CS Student Outcomes	C3100	C3200	C2560	C3700	C4400	C1600	C1650	C2600	C3600	C1900	C4102	M2600
CSSO-1	X	X	X	X						X		X
CSSO-2					X			X	X			
CSSO-3					X	X						
CSSO-4					X						X	
CSSO-5					X				X			
CSSO-6					X				X			
CSSO-7		X		X	X	X	X	X	X			
CSSO-8							X	X				

Prerequisites are chosen to assure students have the appropriate skills and knowledge to succeed in each key course. For example, the prerequisite for CSC 1650 (Digital Electronics) is CSC 1600 (Introduction to Electronics). The prerequisite assures that students have the electrical knowledge and laboratory skills to be effective in CSC 1650.

The chart on the next page shows the prerequisite structure.

CS Major Required Course Prerequisites Flowchart



*Additional prerequisites:

CSC1500: Passing the math placement exam or MATH0200 for CSC1500
CSC1600: MATH1300
CSC1900: MATH1250 or 1300 or equivalent knowledge
CSC3710: CSC2400
CSC2560: CSC1900 or Math1900
CSC4005: Prerequisite/Concurrent: CSC 3450
CSC4102: Junior/Senior status
CSC4350: CSC1550
CSC4600: MATH2600

Other requirements:

MATH1300, MATH1800, MATH2300, MATH2400, MATH2600, PHYS2300, PHYS2400, SPCH1000

Four CSC electives at or above 3000 level.

The courses that cover each curricular area are listed below. Of the courses listed below, those marked with an asterisk indicate elective courses. Others are required.

General Criteria

- Techniques, skills, and tools necessary for computing practice. Students learn modern software development tools from the beginning. The development environments include NetBeans, Visual Studio, Anaconda, IDLE, MPLAB-X, Android Studio, Unity, etc. CASE tools include Visio and Dia. Electrical circuits are simulated using MultiSim, Logisim, EDAplayground (online) and Xilinx WebPack. Skills include algorithm and software design, programming skills, analytical skill, problem solving based on deconstruction and interative refinement, circuit construction and the use of electronic test instruments, among others.

- Principles and practices for secure computing. Secure computing is emphasized in discrete math (CSC/MATH 1900), algorithms (CSC 3700), Web Programming (CSC 3050*), Data Modeling (CSC 3011), Ethical Hacking (CSC 4005*) and Computer and Network Security (CSC 4350*). Topics include: encryption/decryption algorithms, cryptoanalysis, DES, AES, RSA, various vulnerability detection and exploitation methods, etc.

- Local and global impacts of computing solutions on individuals, organizations, and society. Various classes present a historical perspective that demontrates the impact of computing solutions on society. However, Ethical Issues in Computer Science (CSC 4102) and Software Engineering (CSC 4400) have most substantial coverage.

Program Criteria

Courses that cover the areas in the program criteria are listed below. The courses marked with an asterisk are electives. A detailed compilation of the topics and hours of coverage is available in a separate spreadsheet. The coverage is modeled after the ACM Computing Curriculum 2013 guidelines. A table indicating the major topics and the courses that cover them is in Appendix A with our course syllabi. The courses below address substantial coverage of the areas specified in Criteria 3.

Curricular Area	**Course Coverage**
Algorithms and Complexity	CSC 3700, CSC 1500, CSC 1550, CSC 2560, CSC 3200, CSC 4400
Computer Science Theory	CSC 1900, CSC 3700, CSC 1650
Concepts of Programming Languages	CSC 1500, CSC 1550, CSC 2560, CSC 3200, CSC 3300*, CSC 3004*
Software Development	CSC 4400, CSC 3011, CSC 3050*, CSC 3350*
General-Purpose Programming Languages (substantial coverage)	CSC 1500, CSC 1550, CSC 2560, CSC 3500*
Computer Architecture and Organization	CSC 2600, CSC 3600, CSC 3350*
Information Management	CSC 3011, CSC 3050*, CSC 4550*
Networking and Communication	CSC 3600, CSC 3400*, CSC 3450*
Operating Systems	CSC 3100
Parallel and Distributed Computing	CSC 3100, CSC 3004*
Computing-based systems at varying levels of abstraction	CSC 1550, CSC 2560, CSC 3600, CSC 3350*
A major project that requires integration and application of knowledge and skills acquired in earlier course work	CSC 4400, CSC 3011, CSC 3050*, CSC 4550*
Mathematics: at least 15 semester credit hours	CSC/MATH 1900, MATH 1800, MATH 2300, MATH 2400, MATH 2600
Six semester credit hours (or equivalent) in natural science course work intended for science and engineering majors	CSC 1600, PHYS 2300, PHYS 2400

We do not have a coop program.

Materials for each course will be on display at the time of the site visit. These include textbooks (when used) and course binders that contain syllabi, assignments, tests, handouts, and sample copies of student work.

B. Course Syllabi

See Appendix A.

CRITERION 6. FACULTY

A. Faculty Qualifications

All full-time faculty members have terminal degrees in Computer Science or a closely related field. Five have doctoral degrees and one is an Academy Engineer from Denmark. Four full-time faculty members have industry experience. All adjunct faculty are either currently working in industry or have extensive industry experience. Table 6-1 can be found on a page that follows. The resumes are included in Appendix B.

B. Faculty Workload

Most faculty members teach in both CS and CIS programs. Thus, the table reflects the proportion of time spent teaching CS versus CIS majors. Some courses are evenly populated with CS and CIS majors while others are predominantly occupied by students of one particular major. In the table below (6-2), the column labeled "percentage of time devoted to the program" considers these commitments as well as graduate teaching, advising, scholarship and administrative responsibilities.

The faculty contract provides for a teaching load of 12 credit-hours per semester, in addition to student advising and administrative duties. Course releases are available for several specific duties and some ad-hoc ones. For example, the Graduate Program chair gets one course release per year in return for scheduling courses and advising graduate students.

Undergraduate advisees are assigned to faculty by the department chair. Each faculty member has between twenty-five and thirty-five advisees. The faculty contract specifies that advising effectiveness is a criterion for faculty evaluation for promotion and tenure, along with teaching and research. The academic calendar includes a three-week period in each semester for academic advising during which students are expected to meet with their advisors to plan their schedules for the following semester. During this period all faculty have extended office hours, and establish appointment campaigns using the SSC (Student Success Collaborative) system. To motivate students to seek a meeting, registration passwords for online registration change each semester, and the new passwords are distributed to advisors before the advising period.

The department reserves three hours of dedicated time each week to discuss administrative issues, student awards, and the academic programs. Since our last self-study we have effectively used this time to plan new concentrations and related courses, perform faculty search activities, and update the curriculum.

C. Faculty Size

The Department is in the process of hiring another full time faculty member. The candidate will hold a terminal degree in Computer Sciences and support newly appointed coordinators for each the Computer Sciences and Computer Information Systems programs. The department faculty has been steadfast in support of students and vigilant in its oversight and management of the CS program. The department takes pride in reporting that there has been no situation when a CS or CIS student's graduation was delayed because of problems in course offerings. Equally significant, the department has a history of having faculty members prepared to teach most required courses.

Looking forward, the proposed faculty hire will provide needed support for workload gaps involving faculty members who may be seeking sabbaticals within the next 4 years.

D. Professional Development

The university policy is to schedule one scholarship day in each week for each faculty member, during which there are no classroom responsibilities.

Every faculty member in the university is entitled to an annual professional development payment, which may go toward tuition, software, hardware, travel, or other professional development expenses. The amount is fixed annually; usually it is between $700 and $900. For 2018-2019 the amount was $832.

There is a university-wide procedure, whereby faculty can request the release of one course or a three-day schedule for research purposes.

The department has a small travel budget. Most faculty members attend at least one professional meeting in the year. Academic Affairs provides additional funds such as special project grants for faculty research and travel.

The professional development activities for each faculty member is shown in the table below.

Faculty Name	**Professional Development Activities**
Kevin Austin	• Microchip Webinars: Advanced Arduino Debugging (April 10, 2019) and Developing with AVR in MPLAB-X (March 26, 2019) • Xilinx Webinars: ARM processors on Xilinx FPGAs. (January 24, 2019) and No hardware experience? No problem! Xilinx MicroBlaze processors are for everyone. (November 5, 2018) • Computer vision robotics / IoT project. Developing a system for mobile robot tasking / game-playing based on a single video camera system broadcasting JSON objects to robotic platforms through an MQTT server.

	• Media production for education. Experimenting with video presentation techniques for classroom demonstrations. • Collaborated with Dr. Mahadev on "Sound Localization for Robots" and presented at CCSE in Austin, TX. • Bat echolocation signal processing. My field research has yielded a large library of bat echolocation calls. Extracting significant echoes from is a challenging continuing research activity.
Brady Chen	• Northeast OER Summit, Amherst, MA (May 22-23, 2019) • Massachusetts STEM Summit 2018, November 14, 2018, DCU Center, Worcester, MA • National Initiative for Cybersecurity Education (NICE) 2018 Conference and Expo, November 6-7, 2018, Hyatt Regency Miami, Miami, Florida. • ABET Institute for the Development of Excellence in Assessment Leadership (IDEAL), Baltimore, July 31- August 3, 2017 • 2016 Reconnect Workshop, Mathematical and Computational Tools of Cybersecurity, West Point, NY, June 12-18, 2016. • SIAM Conference on Applied Linear Algebra, Atlanta, GA, October 26-30, 2015. Presented Topic: Divide-and-Conquer Algorithm for Computing the Moore-Penrose Inverses • 2015 NSF National Workshop on Teaching an Undergraduate Parallel Programming Course with Pattern Programming in Washington, DC, July 12 –13, 2015.
Natasha Kurtonina	• Developed all materials for undergraduate course "Game Programming" • Updated course material for "Programming Languages" • Developed all materials for a graduate class "Machine Learning" • Conferences and seminars o Harvard Logic Seminar (normally meets every other week) o UConn Logic Seminar (visit twice every semester) o American Association of Symbolic Logic. 2018 Winter Meeting Savannah Convention Center Savannah, Georgia January 3–5, 2018 o American Association of Symbolic Logic. 2017 Winter Meeting Atlanta, GA, USA January 6–7, 2017
Frits Lander	• ABET Symposium, April 19-21, 2012, St. Louis, MO • ABET Symposium, April 14-16, 2011, Indianapolis, IN
Nadimpalli Mahadev	Attended workshops and conferences on Cybersecurity such as the SEED, NICE, CReST etc. to learn more about this emerging field. As a result, he proposed to the university administration introducing of a cyber security concentration as a modest first step. Since then, the cyber security concentration was implemented in our CIS program and the enrollments are beginning to show its popularity. Developed ethical hacking courses as part of this concentration, created a hacking lab that is not connected to university network system. The course is offered during the last two Spring semesters by Dr. Mahadev.

	• Attended ITiCSE 15 where he met with colleagues of similar interest in exploring the idea of holistic approach to teaching computer concepts through research papers. As a result, a group of us experimented teaching at least a course using this approach. Used this approach in graduate Design and Analysis of Algorithms where each student made two or more presentations explain the results of a published paper in a peer-reviewed journal. Each paper was hand-picked by the instructor and contains explanation of an algorithm to solve a hardware, networking or operating systems problem and discusses its efficiency. The course went well and may be repeated again. The team presented a paper at ITiCSE 17 explaining possible implementations and reporting their results on this topic. • Collaborated with Dr. Austin on "Sound Localization for Robots" and presented at CCSE in Austin, TX. • Attended a workshop related to test driven development in software engineering.
Ricky Sethi	□ Interview, Future of Artificial Intelligence, Newsy Television, 2018. https://www.youtube.com/watch?v=VU-fVl0pqNM □ Keynote Speaker, Fact-Checking via Structured Discussions in Virtual Communities, 3rd International Workshop on Social Media World Sensors (Sideways), Prague, Czech Republic, 2017. □ Editorial Board Member, International Journal of Computer Vision & Signal Processing, 2011 - Present

E. Authority and Responsibility of Faculty

Our department has a system of course "ownership", under which one faculty member is primarily responsible for each course, although others may teach it. The content of the course would not change without the oversight of its "owner". In addition, the department discusses the goals for most of the courses in the program, and from time to time we adjust the goals for individual courses to better meet the program objectives. For example, in the last two years, we have been experimenting with teaching CS I using the Python language instead of or in addition to Java. This is a change in content, but it is not a change in goal or direction for the course.

We endeavor to evaluate many of our courses at least biennially. The responsible faculty member will prepare a report and present it at a department meeting. The faculty member reports on how well the course seems to be meeting its goals; the rest of the department shares responsibility for setting the goals, which include measurable student outcomes.
Obviously, the input of the person actually teaching the course has the biggest weight in these decisions.

New courses are usually created by individuals; the idea is worked out, presented to the department; often tested with a trial offering. Before a course can be placed in the university catalog, it must go through the (university-wide) faculty governance system. After it is approved by the department, a formal proposal for the new course must be presented to the university curriculum committee for approval, and finally endorsed by the All University Committee.

For example, we recently introduced a required course in data modeling that will replace our database design course that was only required of CIS students. This change was initiated to have a single course to support the outcomes of both CS and CIS programs. The proposal was discussed by the department curriculum committee with some revisions and then finally approved. It was then submitted to the AUC for final approval and inclusion in our programs.

Table 6-1. Faculty Qualifications

Computer Science Program

Faculty Name	Highest Degree Earned- Field and Year	Rank [1]	Type of Academic Appointment [2] T, TT. NTT	FT or PT [3]	Years of Experience			Professional Registration/ Certification	Level of Activity [4] H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summer work in industry
Kevin Austin	Ph.D. Biomedical Engineering 1987	P	T	FT	10	21	19		M	H	L
Brady Chen	Ph.D. Applied Mathematics 1995	P	T	FT	6	21	17		M	H	L
Natasha Kurtonina	Ph.D. Computer Science 1995	ASC	T	FT		21	19		M	H	L
Frits Lander	Academy Engineer Mechanical Engineering 1968	ASC	T	FT	14	38	37		L	H	M
Nadimpalli Mahadev	Ph.D. Combinatorics and Optimization (CS) 1984	P	T	FT		35	20		L	H	L
Ricky Sethi	Ph.D. Computer Science 2009	ASC	TT	FT	3	10	6		M	H	L
Brian MacKay	Ph.D. Computer Science & Engineering 1996	A	NTT	PT	15	2	1		L	L	L
Orlando Montalvo	M.S. Computer Science	A	NTT	PT		6	6		L	L	H

Instructions: Complete table for each member of the faculty in the program. Add additional rows or use additional sheets if necessary. Updated information is to be provided at the time of the visit.

1. Code: P = Professor ASC = Associate Professor AST = Assistant Professor I = Instructor A = Adjunct O = Other
2. Code: TT = Tenure Track T = Tenured NTT = Non Tenure Track
3. At the institution
4. The level of activity, high, medium or low, should reflect an average over the year prior to the visit plus the two previous years.

Table 6-2. Faculty Workload Summary

Computer Science Program

Faculty Member (name)	PT or FT[1]	Classes Taught (Course No./Credit Hrs.) Term and Year[2]	Program Activity Distribution[3]			% of Time Devoted to the Program[5]
			Teaching	Research or Scholarship	Other[4]	
Kevin Austin	FT	Digital Electronics Section 1 (CSC 1650/4 credits) Spring 2019 Digital Electronics Section 2 (CSC 1650) Spring 2019 Digital Electronics Section 3 (CSC 1650) Spring 2019 Computer Graphics Programming (CSC 4210/3 credits) Spring 2019 Introduction to Electronics Section 1 (CSC 1600) Fall 2018 Introduction to Electronics Section 2 (CSC 1600/4 credits) Fall 2018 Introduction to Electronics Section 3 (CSC 1600/4 credits) Fall 2018 Introduction to Electronics Section 4 (CSC 1600) Fall 2018 Introduction to Electronics Section 5 (CSC 1600) Fall 2018	70%	20%	10%	90%
Brady Chen	FT	Computer Science I Section 3 (CSC 1500/3 credits) Spring 2019 Computer Science I Section 4 (CSC 1500/3 credits) Spring 2019 Systems Design and Implementation (CSC 4700/3 credits) Spring 2019 Operating Systems (CSC 3100/3 credits) Fall 2018 Computer and Network Security (CSC 4350/3 credits) Fall 2018	50%	20%	30%	70%
Natasha Kurtonina	FT	Discrete Mathematics Section 1 (CSC 1900/3 credits) Spring 2019 Discrete Mathematics Section 2 (CSC 1900/3 credits) Spring 2019 Systems Programming (CSC 2560/3 credits) Spring 2019 Algorithms & Data Structure (CSC 3700/3 credits) Spring 2019 Systems Programming Section 1 (CSC 2560/3 credits) Fall 2018 Systems Programming Section 2 (CSC 2560/3 credits) Fall 2018 Programming Languages (CSC 3200/3 credits) Fall 2018 Game Programming (CSC 3650/3 credits) Fall 2018	70%	25%	5%	50%
Frits Lander	FT	Introduction to Programming (CSC 1000/3 credits) Spring 2019 Computer Applications Section 3 (CSC 1100/3 credits) Spring 2019	60%	5%	35%	20%

		Computer Applications Section 4 (CSC 1100/3 credits) Spring 2019 Business Programming (CSC 2700/3 credits) Spring 2019 Introduction to Programming (CSC 1000/3 credits) Fall 2018 Computer Applications Section 1 (CSC 1100/3 credits) Fall 2018 Computer Information Systems (CSC 1400/3 credits) Fall 2018 Ethical Issues in Computer Science (CSC 4102/1 credits) Fall 2018				
N. Mahadev	FT	Computer Science II Section 3 (CSC 1550/3 credits) Spring 2019 Computer Science II Section 4 (CSC 1550/3 credits) Spring 2019 Ethical Hacking (CSC 4005/3 credits) Spring 2019 Software Engineering (CSC 4400/3 credits) Spring 2019 Computer Science I Section 3 (CSC 1500/3 credits) Fall 2018 Computer Science I Section 4 (CSC 1500/3 credits) Fall 2018 Computer Science II Section 1 (CSC 1550/3 credits) Fall 2018 Computer Science II Section 2 (CSC 1550/3 credits) Fall 2018 Web Programming with Java (CSC 3050/3 credits) Fall 2018	70%	20%	10%	60%
Ricky Sethi	FT	Topics: Computational Thinking (CSC1002/3 credits) Spring 2019 Computer Science II Section 1 (CSC 1550/3 credits) Spring 2019 Computer Science II Section 2 (CSC 1550/3 credits) Spring 2019 Topics: Data Science (CSC 3005/3 credits) Spring 2019 Topics: Computational Thinking (CSC1002/3 credits) Fall 2018 Computer Science I Section 1 (CSC 1500/3 credits) Fall 2018 Computer Science I Section 2 (CSC 1500/3 credits) Fall 2018 Computer Science I Section 6 (CSC 1500/3 credits) Fall 2018 Database Systems (CSC 2400/3 credits) Fall 2018	50%	40%	10%	50%
Brian MacKay	PT	Computer Science I Section 1 (CSC 1500/3 credits) Spring 2019 Computer Science I Section 2 (CSC 1500/3 credits) Spring 2019 Microprocessors (CSC 3600/4 credits) Spring 2019 Computer Organization Section 1 (CSC 2600/4 credits) Fall 2018 Computer Organization Section 2 (CSC 2600) Fall 2018 Computer Organization Section 3 (CSC 2600) Fall 2018	60%	0%	40%	60%
Orlando Montalvo	PT	Local Area Networks (CSC 3450/3 credits) Spring 2019 Data Communications & Networking (CSC 3400/3 credits) Fall 2018	50%	0%	50%	50%

1. FT = Full Time Faculty or PT = Part Time Faculty, at the institution
2. For the academic year for which the Self-Study Report is being prepared.
3. Program activity distribution should be in percent of effort in the program and should total 100%.
4. Indicate sabbatical leave, etc., under "Other."
5. Out of the total time employed at the institution.

CRITERION 7. FACILITIES[1]

A. Offices, Classrooms and Laboratories

The department of Computer Science is housed in Edgerly Hall, the oldest on campus. Built as a grade-school, and taken over as a campus school by the normal school that was the beginnings of Fitchburg State. It is not as grand as the college buildings built soon after it, but it has good bones and has been well-maintained.

Six years ago, the third floor of the building has been completely remodeled, and the department and faculty offices have been moved there. The new space is bright and airy, and several faculty members' new offices will be bigger. The department secretary shares airspace with the Math department secretary in an open but windowless office/corridor, which gives both students and faculty easy access to her.

In 2018, the building has been renovated again and a new elevator has been installed, making the building accessible to disabled people.

The department uses two computer laboratories as classroom space, and another classroom has some computers for use in networking labs. Two additional classrooms serve as computer hardware and electronics laboratories. Almost all of our classes are held in Edgerly Hall, where the department offices and faculty offices are located.

The classrooms are well-sized for the 20-25 student classes we strive to teach, and the acoustics are excellent. All of our classrooms and laboratories have lecture podiums and projection systems. In two of the labs, the student computers are thin clients, with the file storage and some of the computing taking place remotely; in labs where hardware and networking are taught, we have more traditional machines.

Classrooms and labs are available outside class hours. Computing facilities are nearly always available. Tools like oscilloscopes are locked in cabinets, for which several faculty members and the department office have keys, so they are usually available only during the day.

B. Computing Resources

Fitchburg State recommends that all students have a mobile device, either a laptop or a tablet. While there are numerous open labs available on campus, the ability to do work where and when you need to means that a mobile device makes your academic life easier. Students are directed to Student Computer Recommendations (https://www.fitchburgstate.edu/offices- services-directory/technology/stucomp/) for help in deciding which device is right for them.

Fitchburg State University provides computer resources that are available to students outside of classroom hours. The first floor of Hammond Hall and McKay C-163 house open computer labs, offering a convenient place to study and do research. Edgerly Hall and Conlon Hall house department-specific labs that provide resources to meet the unique needs of students and faculty in those academic programs. Since 2008, all entering students will complete computer literacy requirements as defined by departments.

An electronic printing system allows students to print from a mobile device and student computer labs. Students need to use their OneCard (https://www.fitchburgstate.edu/offices-services-directory/onecard/) and Falcon Key (https://www.fitchburgstate.edu/offices- services-directory/technology/falconkey/) to use the service.

Additionally, six Ricoh multifunction devices installed across campus exist for students' use. These devices are located in McKay C163 Business Lab, Conlon Bridge between Fine Arts and Industrial Technology, Hammond Library first floor (three available) and Antonucci Science Complex (1st floor lounge).

A 24/7 Help Desk/Call Center offers technical support to students, faculty and staff by phone at (978) 665-4500 or toll-free at 866-830-0518. Users can also submit a ticket by completing the Online Help Desk Request Form (http://helpdesk.fitchburgstate.edu/). There is also walk- in service, during normal business hours, available at the on-campus Help Desk in Conlon Hall, Room 236. View our Hours of Operation (https://www.fitchburgstate.edu/offices- services-directory/technology/help/) page for a schedule of walk-in hours.

The Fitchburg State Help Desk will make a reasonable effort to repair computers owned by students who currently attend the university. Laptop loaners are available to students whose laptops are left at the Help Desk for extended periods of repair time.

All classroom spaces are equipped with a ceiling-mounted projector, Windows computer for instructor use, speakers, Apple TV and a Blu-ray player. Document cameras are also available in many classrooms, and select classrooms have a SMART Board. Other resources such as software for lecture capture and conferencing are also available. Details can be found in the Classroom Technology guide (https://www.fitchburgstate.edu/uploads/files/Technology/classroom_technology_guide.pdf).

The Fitchburg State University campus network provides high-speed access to the internet, including access to Internet2. Access to the network is available throughout campus, including faculty and staff offices and departments, computer labs, study areas, residence halls, and in a number of classrooms. Wi-Fi is also available in all the buildings on campus, including green spaces. The Fitchburg State University wireless network conforms to current 802.11 standards operating at speeds up to 100 Mbps. All wired connections offer 1Gbps speeds. Secure remote access to the campus network is also available through our VPN service.

All computer laptops and desktops issued to full time faculty and staff by the university are refreshed periodically to ensure the most efficient and reliable use of these systems. In line with manufacturer's warrantees, Fitchburg State University replaces laptop computers every three years and desktop computer every four years. This refresh schedule is based on the fiscal year. Maintenance and upgrading of classroom software and software used in various labs is handled by Information Technology in coordination with faculty needs.

C. Guidance

Primary source of information for students regarding the use of various hardware and software resources is the classroom experience. Instructors provide hands-on training when the resources are introduced to the students for the first time. These include the following:

- Email: The University provides an email account to all the students, faculty and staff, which is the main source of communications for all the university-related work.
- Blackboard: This web tool, which is widely used across many colleges, is introduced to the students in classrooms. The instructors can post syllabus, handouts, assignments, tests, students grades etc. to the blackboard and registered students can access them, submit assignments, view grades, view instructor comments on graded assignments, tests etc.
- Hardware: In hardware courses such as Intro to Electronics, Digital Electronics, Computer Org, Microprocessors, Data Communications and Local Area Networks students are introduced to many hardware tools, components and software tools such as multimeter, oscilloscope, simulation software, etc.
- Software: Various programming languages and the associated IDEs, Databases, web servers, telnet and other software that are used in various courses are introduced in the classroom.
- J Drive: The University provides all the faculty and students with a network drive to store individual data and files. Students can access the J Drive from any of the campus computer labs as well as from dormitories. Faculty can access them online from anywhere.
- I Drive: Some departments may request a shared network drive for the strict use of their own faculty members. The computer science department has one. The faculty can access them online or from campus.
- Web Site: The University provides online account for interested students to host their own individual web pages. Students were introduced to this tool in web development courses.
- Screen capture video: the University provides software for faculty to produce screen- capture video either during a classroom experience or elsewhere (Screencast-o-matic) and maintains a web site so students may access the content.

Another source of information for other computer resources is the advising period. Every student is assigned a faculty member from their own major as an advisor and during the advising period, students meet with the advisor regarding the study plan for the next semester and beyond. A major tool introduced at this time is the banner system (also known as Web4). During academic advising, faculty members show their advisees how to complete various registration-related tasks using Web4 such as:

- Viewing the schedule of course offerings for the following semester
- Registering for classes online.
- Obtaining a degree evaluation which shows the courses a student needs to complete the program.
- Viewing an unofficial transcript.
- Checking final grades.
- Viewing class schedule.

Finally, the Information Technology's web site provides a lot of useful information for both the students and faculty members regarding various resources available for their computing needs. These include information on:

- The list of university owned campus-wide site licenses.
- Free downloads of anti-virus software and Microsoft Office and Windows OS.
- Software available for free download specifically to Computer Science faculty and students.
- Helpdesk for maintenance and troubleshooting of computer resources.
- Servicing of faculty/student laptops.
- Printing facilities for the faculty and students.
- Virtual desktop systems (thin clients) available across campus.
- Various computer labs and hours they are open. Some of these are exclusively for the use of Computer Science faculty and students, as required by the license agreements.
- OneCard system for purchase of discounted software etc.
- Computer refresh schedule for the faculty (more on it in the next section).
- Policies and procedures governing the use of college-owned hardware and software.

D. Maintenance and Upgrading of Facilities

Maintenance and upgrading of the technology-related facilities is the responsibility of the Information Technology division. Here are some procedures used in this regard.

- Helpdesk is a call center used by students and faculty alike to report any issues with their accounts, and the electronic equipment provided in the classrooms, labs, library and other locations, and with various web tools, phone systems etc.
- All computer desktops and laptops provided by the university are refreshed periodically to ensure the most efficient and reliable use of these tools. In line with manufacturer's warrantees, Fitchburg State University replaces desktop computers every four years and laptop computers every three years. This refresh schedule is based on the fiscal year and individual systems will be refreshed after lab systems are upgraded.
- Maintain all the upgrades of web tools and classroom software such as the Blackboard, Banner system, software used in various labs etc. These upgrades usually take place between semesters to refresh the systems for the new academic year.
- Data backup on regular basis of all Blackboard accounts, the Banner system, and the faculty/staff email accounts, and the network drives etc. The university maintains all the Blackboard course accounts for up to four years. Some student records are removed when the student graduates.

E. Library Services

The complete review of library services is provided in Appendix F.

F. Overall Comments on Facilities

Fitchburg State University has various departments that work collaboratively to ensure that the university complies with workplace safety, fire and life safety codes and standards. The department includes the Capital Planning and Maintenance, Housing and Residential Services, Environmental Health and Safety, and Risk Management, as well as University police. All buildings meet or exceed local and national fire safety codes and building codes and are equipped with modern fire alarm systems including smoke detectors and sprinkler systems. These systems are routinely inspected by the City's Fire Department, State Building Inspectors and also specialized contractors.

[1]Include information concerning facilities at all sites where program courses are delivered.

CRITERION 8. INSTITUTIONAL SUPPORT

A. Leadership

The department chair is in charge of course scheduling, budget requests, and a myriad of administrative detail. Over the past six years, the department chairperson has been Dr. Brady Chen. The Computer Science program is a central part of our departmental mission.

The primary support points for the CS program are course scheduling and staffing. Secondary needs include recruiting and advising. Professor Frits Lander has been very active in new student forums and new student advising. The university pays the department chair a small summer stipend, and during the summer there are several orientation meetings for new students, at which there is always someone present from our department.

B. Program Budget and Financial Support

The annual budget is submitted each spring. Our department budget includes funds for support of courses, for supplies, etc. It does not include salaries for faculty and staff, which are part of another process.

We do not use graders or teaching assistants in our department. In years past, the department hired lab monitors to oversee our open computer labs, but this budget line has been moved to the Information Technology Office, along with responsibility for maintaining and upgrading the labs themselves.

The technologies we principally rely upon to deliver instruction are 1) digital projectors in the classrooms; 2) computer facilities in the classrooms and laboratories, including specialized computer systems for teaching network administration and microprocessor development; 3) electronic tools and parts. Of these, items (1) and (2) are now supported and maintained by the IT department. This is slightly clumsy for our specialized systems. On the one hand, IT would like to standardize all computers on campus, which would clearly eliminate their support for our special needs. A part-time technical person to attend to our special needs is desirable.

Electronic tools like oscilloscopes and student breadboard systems are still part of the department budget; these are used by courses in the Computer Science program. Maintenance is performed by the faculty using the equipment.

C. Staffing

Our current department administrative assistant is Christine Nickoloff who was hired in 2015 to replace Ann Larsen. Ann left the job and pursued her career in math department. Christine has proven to be an excellent addition to the department with both a willingness to take on new roles and expand her responsibilities, and with strength in carrying through on her existing duties.

The university has a series of workshops on support software and the academic/budget database systems for administrative personnel. There are good benefits, but salaries seem low.

D. Faculty Hiring and Retention

The recruitment and hiring of new faculty is a shared process, which includes faculty members, department chair, and the office of Academic Affairs. The process is defined in the MSCA contract and supported by the Human Resources Hiring Handbook. Human Resources also provides training in the hiring process and ensures that the search has been conducted properly. See (https://www.fitchburgstate.edu/uploads/files/Human_Resources/Faculty%20Search%20and%20Hiring%20Guide%20Updated%20January%202018.pdf). The process begins with requests for new positions from academic department chairs to the Vice President for Academic Affairs. The Vice President reviews these requests and makes recommendations to the President. If the position is approved, a search committee of faculty/librarians complete hiring training, review all application materials, contact candidates, schedule interviews, conduct phone/Zoom and on-campus interviews, and request and communicate administrative details for the hiring process (parking passes, meal vouchers, and travel expense reimbursement) to the candidate. At the completion of a search process, the Search Committee submits its final three candidates unranked to the department chairperson, dean (or library director), who then submits the committee's recommendation and a separate chair, dean (or director) recommendation to the Vice President for Academic Affairs. The VPAA reviews the recommendations and makes a separate recommendation to the President, who then makes the final decision and sends the letter of appointment.

In the last ABET visit, the evaluation team identified the weakness in Criteria 6 Faculty. That weakness was addressed by the hiring of Dr. Ricky Sethi. The Department is currently conducting a search for an additional tenure track faculty to support both the Computer Sciences and Computer Information Systems programs. The new position will provide sufficient resources to assign separate coordinators for each the CS and CIS programs.

Of our current six faculty members in the department, the most recently hired will begin his fifth year in the fall of 2019, and the other five have longer terms of service, ranging up to Professor Frits Lander's 37 years. Clearly we don't have a big problem with turnover. We are fortunate that our department has been able to work together very well.

E. Support of Faculty Professional Development

Currently, the institution supports faculty scholarship, research, and creative activity in several ways. Most notably, faculty members have a designated weekly research day. Support for professional development is contractually mandated. Tenured and tenure-track faculty and librarians having served six years at the University are eligible for sabbatical leave. Sabbatical leave is governed by Article XV in the current MSCA collective bargaining agreement. Article XIV calls for professional development funding using a formula for computation enumerated in the agreement. In each academic year, funds are disbursed as a one-time Continuing Scholarship stipend, for which faculty and librarians may apply by articulating the manner in which the funds will be spent. These funds are intended to suit the needs of individual faculty members and are not restricted or limited to certain activities.

Many faculty members use these funds to attend conferences, for subscriptions, membership in professional organizations, and to build personal libraries of materials, which thereby enrich their own personal teaching. Funds are available to all faculty and librarians.
Approximately 90% of the faculty members apply for these funds. Any leftover monies are then redistributed among the original applicants.

The Vice President for Academic Affairs also provides additional professional development support. Each department receives a budget for Travel Funds, which is distributed according to departmental policies. The Vice President for Academic Affairs also has established special grants to support faculty research/creative activity, assessment projects, academic programming to enhance diverse/global perspective, and faculty travel with priority given to tenure track faculty presenting at peer-reviewed conferences. The participation of the faculty in national and international conferences as presenters is encouraged. For a small state university the involvement of the faculty in such professional activities is impressive and well documented in the Fitchburg State Today and other publications.

Other professional development support has been internally created. Faculty and librarians find professional development opportunities through the University's Center for Teaching & Learning (CTL). The CTL was launched in 2006 with the purpose of supporting faculty development through peer-facilitated workshops, presentations, mentoring, institutional monetary grants, and training. One or two full-time faculty receive workload reductions to serve as director(s) for the CTL, and it maintains an Advisory Board comprised of full-time faculty to aid in the development of programs, review applications for the CTL Innovation Grants, and coordinate a mentoring program for new faculty. Faculty professional development is also provided through workshops presented by the Distance Education Coordinator. Approximately 40 face to face sessions are offered annually with audio and video versions of the presentations available as well. GCE hosts a biennial faculty meeting which includes professional development sessions. The last gathering included such workshop topics such as innovative practices in online learning, developing effective rubrics and the challenges of grading. The Harrod Lecture series provides an additional forum for faculty to present their research to the University community. Faculty may also receive support for grant applications through the University's Grant Center, as well as the Office of Academic Affairs when appropriate. In addition, the Grant Center oversees the Ruth Butler Award, (http://www.fitchburgstate.edu/offices/grants-research/grant-center/ruth-butler- grants/), created to support professional or artistic development, or activities that are appropriate to furthering the goals of Fitchburg State University by full-time members of the faculty, librarians, administrators, and students of Fitchburg State.

Faculty performance is also rewarded through two yearly awards offered by the University. One, the Mara Award, is given to a professor for Excellence in Teaching; the second is the Faculty Award for Research and Scholarship. In addition, in 2010 in conjunction with the second annual graduate commencement ceremonies, an award was established to recognize excellence and contributions for graduate programs. In all cases, funds which support professional development directly benefit teaching, since they provide access to the most current and well-researched course material.

Additional scholarship, research and creative activities are currently supported through several initiatives including the Regional Economic Development Institute (REDi). The REDi's Advisory Board, comprised of North Central Massachusetts leaders from businesses, municipalities, community organizations, media, and university department chairs, supports REDi research and outreach regionally. The REDi provides a research office and conference room (space) for faculty, with student assistance as interns, to perform on-campus and off-campus research in areas of economic development for North Central Massachusetts. Two distinct studies were completed in the first year of operation and a regional economic development summit co-sponsored by Fitchburg State University with the Massachusetts Lieutenant-Governor and Mount Wachusett Community College. Four more studies are underway, inclusive of one focused on regional broadband advancement and coordinated with the Massachusetts Governor's Office, with numerous public forums held and several studies published during REDi's second year.

PROGRAM CRITERIA

Our program is modeled after the ACM Computing Curriculum 2013 guidelines. How the program satisfies the program criteria is described in Criterion 5, section A, item 5. In the area of curriculum, the criteria are satisfied as demonstrated in Table 5-1. Table 5-1 shows the breakdown of program credit-hours based on the general topic areas of math and sciences (38), computing fundamentals (18), advanced computing (31), general education (24) and others (9).

Each course outline listed in the "Course Syllabi" section contains a table showing how each course objective aligns with student outcomes which in turn are mapped to program educational objectives. This mapping is given in a table shown earlier under Criterion 3 section A which is entitled "student outcomes".

As may be seen in Table 6-1, 50% of the faculty members have PhD in areas of Computer Science and others have closely related degrees and/or industry experience.

APPENDICES

APPENDIX A – COURSE SYLLABI

For those courses which may be applied toward the Computer Science major we indicate in the list of topics for the course the number of lecture hours devoted to each subtopic in the 2013 Computing Curricula Body of Knowledge list.

The following table summarizes how required courses relate to core areas (2013 Computing Curricula Body of Knowledge list).

	CS Major Required Courses													
CC 2013 Body of Knowledge vs. required courses	CSC 1500 RS	CSC 1550 NM	CSC 1600 KA	CSC 1650 KA	CSC 1900 NK	CSC 2560 NK	CSC 2600 KA	CSC 3011 RS	CSC 3100 BC	CSC 3200 NK	CSC 3600 KA	CSC 3700 NK	CSC 4102 FL	CSC 4400 NM
AL: Algorithms & Complexity	X	X	X	X		X				X	X	X		
AR: Architecture & Organization			X	X			X		X	X	X			
CN: Computational Science	X		X	X		X		X			X			
DS: Discrete Structures				X	X	X				X		X		
GV: Graphics and Visualization		X												
HCI: Human-Computer Interaction	X	X									X			X
IAS: Information Assurance & Security								X				X		
IM: Information Management								X						X
IS: Intelligent Systems			X					X		X		X		
NC: Networking & Communication				X				X			X			X
OS: Operating Systems						X			X		X			
PBD: Platform-Based Development														
PD: Parallel & Distributed Computing							X	X			X	X		
PL: Programming Languages	X	X				X				X				
SDF: Software Development Fundamentals	X	X				X		X		X	X	X		
SE: Software Engineering	X					X		X			X			X
SF: Systems Fundamentals			X	X			X				X			
SP: Social Issues & Professional Practice	X		X	X		X	X	X			X			X

1. Course number and name
   CSC 1500 Computer Science 1
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Ricky J. Sethi
4. Text book, title, author, and year
   Essential Computational Thinking: Computer Science from Scratch. by Ricky J. Sethi, 2018. Cognella.
   a. other supplemental materials
      Java Foundations by John Lewis. Pearson.
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course introduces Computer Science by using a high-level programming language. Students will be taught to design programs and implement them using object-oriented programming techniques. This course provides a solid background for further studies in Computer Science by preparing students to enroll in the more specialized high-level software courses.
   b. prerequisites or co-requisites
      Passing the Math placement exam or MATH 0200
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1: History and evolution of computers and programming languages.
      CO2: The process of designing, coding, compiling, debugging and executing programs.
      CO3: Basic structure, syntax and semantics of a programming language.
      CO4: Basic programming involving data type declarations, arithmetic expressions, methods and input/output.
      CO5: Using control structures such as if-else, switch, while and for loop.
      CO6: Designing and coding graphical user interfaces with event handling.
      CO7: Creating and using overloaded methods and constructors.
      CO8: Object oriented programming concepts such as encapsulation, inheritance and polymorphism.
      CO9: Working with arrays.
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1								
CO2	Y							
CO3	Y							
CO4	Y					Y		
CO5	Y					Y		
CO6	Y					Y		
CO7	Y					Y		
CO8	Y					Y		
CO9	Y					Y		

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1								
CO2	Y							
CO3	Y							
CO4	Y							
CO5	Y							
CO6	Y							
CO7	Y							
CO8	Y							
CO9	Y							

7. Brief list of topics to be covered
    a. Applications of computers and internet.
    b. Security and ethical issues in the use of computers.
    c. Evolution of computers and programming languages.
    d. The process of designing, coding, compiling, executing and debugging computer programs.
    e. Structure, syntax and semantics of a program. Programming style conventions.
    f. Data type declarations, arithmetic operations and basic input/output.
    g. Control structures.
    h. Classes, objects and constructor overloading.
    i. Creating and using methods.
    j. Scope of variables.
    k. Graphical user interface programming and event handling.
    l. Introduction to arrays.

1. Course number and name
   CSC 1550 Computer Science 2
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Ricky J. Sethi
4. Text book, title, author, and year
   Java Foundations by John Lewis. Pearson. 2013.
   a. other supplemental materials
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course builds on the concepts covered in Computer Science I. Topics covered include inheritance, polymorphism, recursion, advanced GUI programming, exception handling, and input/output handling. Students use an integrated development environment to create, compile, run and debug programs.
   b. prerequisites or co-requisites
      CSC 1500
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1. Data Structures in Java: Arrays and Strings. Simple Sorting and Searching Algorithms. Multidimensional Arrays.
      CO2. HTML and Applet Basics. Event Driven Programming. The Applet Life Cycle. Interactive Applets.
      CO3. How to use int(), start(), stop(), destroy(), paint(), repaint() methods within applets.
      CO4. Graphics Basics. How to use drawString() method. How to use setFont() and setColor() methods. How to set an applet's background color. How to create graphic objects.
      CO5. The concept of Inheritance. How to extend classes. What is a derived class.
      CO6. Public and Private variables and methods. Constructors with and without arguments. How to access Superclass methods that have Constructors.
      CO7. Advanced Inheritance concepts. Abstract Classes and Dynamic Method Binding.
      CO8. Abstract Windows Toolkit.
      CO9. Using Layout Managers and the Event Model.
      CO10. Exception Handling in Java. The concept of Exception. How to throw and catch and exception?
      CO11. Input/Output and File Techniques. File organization and streams. How to use streams. How to write a file and to read data from a file. Writing and reading formatted file data.

b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y					Y		
CO2	Y							
CO3	Y					Y		
CO4	Y					Y		
CO5	Y					Y		
CO6	Y					Y		
CO7	Y					Y		
CO8	Y					Y		
CO9	Y					Y		
CO10	Y					Y		
CO11	Y					Y		

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1	Y							Y
CO2	Y							Y
CO3	Y							Y
CO4	Y							Y
CO5	Y							Y
CO6	Y							Y
CO7	Y							Y
CO8	Y							Y
CO9	Y							Y
CO10	Y							Y
CO11	Y							Y

7. Brief list of topics to be covered
    a. Representing Advanced Data Structures in Java: Arrays and Introduction to Sorting and Searching Algorithms; Linked Lists Representations in Java
    b. Applets
    c. Graphics
    d. Inheritance: Introduction to Inheritance. Superclasses and Subclasses; Advanced Inheritance concepts. Abstract Classes and Dynamic Binding
    e. Understanding the Abstract Windows Toolkit
    f. Exception Handling in Java
    g. Input/Output and File Techniques
    h. Multithreading and Animation

1. Course number and name: CSC 1600: Introduction to Electronics
2. Credits and contact hours: 4 credits / 82.5 hours.
3. Instructor's or course coordinator's name: Kevin Austin
4. Text book, title, author, and year: Floyd, Thomas L. Electronics Fundamentals: Circuits, Devices and Applications. Pearson Prentice Hall, 2010
   a. other supplemental materials: Electronic prototyping, test and measurement equipment and components, robotic chassis, MultiSim circuit simulation software, lab handouts, problem-solving handouts, notes and lectures are posted on Blackboard for student review.
5. Specific course information
   a. brief description of the content of the course (catalog description)

      This course provides an introduction to the electrical signals, circuits and solid-state devices fundamental to understanding the operational characteristics of digital computers. Methods for the theoretical analysis of electrical circuits containing resistive and capacitive elements are covered extensively. Electrical circuit analysis provides a firm scientific foundation for discussion of practical circuits relevant to computer science. Topics covered in the class include: electronic properties of materials, electrical signals, DC circuit analysis, transient circuit analysis, diodes, transistor switching circuits, integrated circuits, operational amplifier applications, digital-to-analog conversion and analog-to-digital conversion. In the laboratory, students build circuits and make electrical measurements using modern laboratory instruments to validate and reinforce the theory presented in class.

   b. prerequisites or co-requisites: MATH 1300 (Pre-calculus)
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☒required; ☐elective; ☐selected elective.
6. Specific goals for the course:
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.

   The purpose of this course is to give students a foundation in the electronic and scientific principles that underlie the operation of computing machinery and to introduce them to the laboratory procedures and equipment that they will subsequently use in Digital Electronics. (CSC1650). Upon completion of the course, a student should be able to do the following:

   - CO1: demonstrate knowledge of fundamental electrical quantities;
   - CO2: analyze DC circuits containing multiple resistors and voltage sources to solve for all branch voltages and currents;
   - CO3: construct and troubleshoot simple electronic circuits containing resistors, capacitors, transistors and operational amplifiers using a solderless breadboard, digital multimeter and oscilloscope;
   - CO4: measure/compute standard metrics for various waveforms particularly pulse trains;
   - CO5: perform transient and steady state analysis of circuits containing resistors and capacitors to understand signal degradation and D/A conversion;
   - CO6: use operational amplifiers as comparators and amplifiers;
   - CO7: analyze switching circuits containing transistors and diodes;
   - CO8: maintain a lab notebook, write a technical report, and orally discuss lab results with a supervisor;

b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

	CS Program Student Outcome							
	1	2	3	4	5	6	7	8
CO1			Y				Y	
CO2								
CO3							Y	
CO4								Y
CO5								Y
CO6								Y
CO7								Y
CO8			Y					

7. Brief list of topics to be covered:

- Why study electronics?
    - Computers are fundamentally electronic devices
    - Understanding electronic principles is essential as computers become smaller, faster and battery-powered
    - Knowledge of electronics is highly valued in any environment where software and hardware are being co-developed
    - The approach to problem-solving using deconstruction, analysis and iterative reconstruction used in this course is generalizable to software development
- Electrical properties of materials: electrons, conductors, insulators, semiconductors
- Electrical concepts:
    - SI units, voltage, current, resistance, conductance, energy, power
    - Power sources, batteries, switches, pull-up, pull-down circuits
    - Electrical signal properties and measurements: period, frequency, amplitude, spectra, rise time, fall time, pulse width, duty cycle, modulation
- Problem-solving: DC circuit theory and analysis
    - Ohm’s law, Kirchhoff’s laws, deconstruction, simplification, iterative reconstruction, circuit simulation, D/A conversion, Thevenin’s theorem
- Systems Point-of-view: Operational amplifier applications
    - Linear: inverting and non-inverting amplifiers, analog computation
    - Non-linear: comparators, analog-to-digital conversion
- RC circuits
    - Capacitance, energy storage: time constant, charge/discharge cycles. Steady-state vs. transient analysis, pulse-width modulation (PWM), D/A conversion
- Semiconductor devices
    - Diodes: P-N junction, diode V-I curve, light-emitting diodes, diode circuits
    - Transistor switching circuits: BJT & MOSFET switches, characteristic curves, current gain, transconductance, load lines, saturation, cutoff, switching circuits, controlling LEDs and motors through PWM

1. Course number and name: CSC 1650: Digital Electronics
2. Credits and contact hours: 4 credits / 82.5 hours.
3. Instructor's or course coordinator's name: Kevin Austin
4. Text book, title, author, and year: "Digital Fundamentals", 10th Edition, by Thomas Floyd ISBN: 978-0-13-235923-8 (Prentice Hall).
    a. other supplemental materials: Electronic prototyping, test and measurement equipment and components, lab handouts, problem-solving handouts, notes and lecture video are posted on Blackboard for student review. MultiSim & LogiSim software, EDAplayground, Vivado and CMOD A7 FPGA module.
5. Specific course information
    a. brief description of the content of the course (catalog description):
    This course provides a study of the fundamental circuit building blocks that are used in the development of digital computers. The theory and practical application of both asynchronous and synchronous electronic logic circuits are covered. Topics included are: binary representations, data transfer methods, error detection and correction, logic gates, logic families, programmable logic devices, Boolean algebraic simplification, Karnaugh maps, combinational logic circuits, adders, comparators, encoders, decoders, multiplexers, demultiplexers, sequential logic circuits, latches, flip-flops, counters, shift registers and memory. Extensive laboratory work supplements the topics studied.
    b. prerequisites or co-requisites: CSC 1600
    c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☒required; ☐elective; ☐selected elective.

6. Specific goals for the course:
    a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic. The purpose of this course is to develop students' understanding of the digital logic principles that are fundamental to the operation of computing machinery. It assumes that students have a knowledge of electrical concepts, electrical signals and laboratory equipment. The course establishes both the theoretical and practical foundation necessary to understand computer organization and microprocessors. Upon completion of the course, a student should be able to do the following:

- CO1: demonstrate how information is represented in a digital computer;
- CO2: demonstrate how to analyze and simplify Boolean expressions;
- CO3: analyze digital logic circuits using truth tables, timing diagrams and state diagrams;
- CO4: develop combinational logic implementations using truth tables and K-maps;
- CO5: demonstrate an understanding of how various functions of combinational logic work;
- CO6: develop realizations of sequential logic using transition tables, K-maps and state diagrams;
- CO7: demonstrate understanding of a digital logic design in a hardware definition language (HDL), simulate the design, transfer it to a programmable logic device (PLD) and verify functionality;
- CO8: maintain a lab notebook and orally discuss lab results with a supervisor;

b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

	CS Program Student Outcomes							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CO1							Y	Y
CO2						Y	Y	Y
CO3						Y	Y	Y
CO4		Y		Y		Y	Y	Y
CO5						Y	Y	Y
CO6		Y		Y		Y	Y	Y
CO7	Y					Y	Y	Y
CO8			Y					

7. Brief list of topics covered:
   - Overview of digital electronics: Examples of digital systems using gates, adders, comparators, decoders, encoders, multiplexers, demultiplexers, counters, shift registers
   - Digital representations
     - Voltages represent logic states: high and low
     - Number systems: decimal, binary, hexadecimal
     - ASCII code, Gray code, error detection and correction codes
     - Serial and parallel data transfer
   - Fundamental gates, Boolean expressions, truth tables and timing diagrams
   - Electrical characteristics of digital logic devices / Guaranteed logic levels and logic device compatibility / Propagation delay
   - Rules and laws of Boolean algebra / Distributive, Commutative and Associative laws
     - DeMorgan's theorem
     - Logic simplification: combinational logic circuit analysis and implementation
     - Truth tables / Standard forms of expressions
     - K-maps: Valid and invalid groupings
   - Functions of combinational logic
     - Adders and comparators
     - Decoders and encoders / Multiplexers and demultiplexers
   - Hardware definition language & programmable logic devices
   - Latches / Gate implementations
     - Gating, edge-triggering
     - Timing diagrams / state diagrams
     - D, T and J-K Flip-flops
   - Sequential circuit analysis and design
     - Registers / counters / shift-registers

1. Course number and name: CSC 1900: Discrete Mathematics
2. Credits and contact hours: 3 credits / 37.5 hours.
3. Instructor's or course coordinator's name: Natasha Kurtonina
4. Text book, title, author, and year:

   Discrete Mathematics and Its Applications, 4th Edition by Kenneth H. Rosen, McGraw-Hill Science/Engineering/Math (December 11, 1998) **ASIN:** B0090TGL1O

   a. other supplemental materials:
      Notes, handouts and lectures are posted on Blackboard for student review.

5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course introduces discrete mathematics as applied to computer science. Topics covered include Boolean logic, elementary set theory, functions, relations, enumeration, proof techniques, number systems and trees.

   b. prerequisites or co-requisites:
      Math1250 or Math 1300 or equivalent knowledge.
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program: ☒required; ☐elective; ☐selected elective.

6. Specific goals for the course:
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The purpose of this course is to develop logical thinking skills and understanding of mathematical applications to computer T. he subject enhances one's ability to reason and ability to present a coherent and mathematically accurate argument. Upon completion of the course, a student should be able to do the following:
      Upon successful completion of this course, a student will have demonstrated knowledge of:

      - CO1: Fundamentals of propositional logic and Boolean expressions analysis
      - CO2: Proof Techniques.
      - CO3: Boolean Logic Techniques in Digital Electronics
      - CO4: Fundamentals of Set Theory
      - CO5: Basic Number Theory
      - CO6: Basic Counting Principles
      - CO7: Basic Graph Theory
      - CO8: Elementary Finite Automata Theory

   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y							
CO2	Y							
CO3	Y							
CO4	Y							
CO5	Y							
CO6	Y							
CO7	Y							
CO8	Y							

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1	Y							
CO2	Y							
CO3	Y							
CO4	Y							
CO5	Y							
CO6	Y							
CO7	Y							
CO8	Y							

7. Major Topics Covered in the Course:
   1. Fundamentals of Propositional Logic
      - Truth Tables
      - Equivalent Boolean Expressions
      - Justification of reasoning patterns
   2. Proof Techniques
      - Proof by Contradiction
      - Mathematical Induction
      - Predicate Logic and Syllogisms
   3. Propositional Logic and Boolean Algebra. Circuits and Gates.
   4. Basics of Set Theory.
      - Set-theoretic Relations and Set-theoretic operations
      - Equivalence of Set-Theoretic Expressions
      - Logic and Set Theory
      - Property of Binary Relations
      - Relations and Functions
   5. Basics of Number Theory
      - Euclid Algorithms
      - Prime numbers and Factorization
      - Modular Arithmetic and Cryptography
      - Summation Principles and Mathematical Induction
   6. Counting Principles, Permutations, Elements of Probability Theory.
   7. Graphs and Trees and Their Representations
   8. Introductions to Models of Computation and Complexity Analysis

1. Course number and name: CSC 2560: Systems Programming
2. Credits and contact hours: 3 credits / 37.5 hours.
3. Instructor's or course coordinator's name: Natasha Kurtonina
4. Text book, title, author, and year:
   C Primer Plus. Stephen Prata. Sams. ISBN: 0672326965
   a. other supplemental materials:
      Notes, handouts and lectures are posted on Blackboard for student review.

5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course provides the student with the fundamentals of structured programming at the systems level. Students obtain a thorough knowledge of the C language, pointers, linked lists, trees and comfort in using the UNIX operating system
   b. prerequisites or co-requisites:
      MATH 1300 (Pre-calculus)
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program: ☒required; ☐elective; ☐selected elective.

6. Specific goals for the course:
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The purpose of this course is to give students a foundation in the structured programming with C and Unix Operating System that underlie the the principles of process generation and process management, dynamic memory management, dynamic Data Structures, including Trees and Linked Lists, that will subsequently be used in Algorithms and Data Structures. (CSC3700). Upon completion of the course, a student should be able to do the following:
      - CO1: Structured programming with C
      - CO2: Dynamic arrays and linked lists
      - CO3: Trees and pointer arithmetic
      - CO4: Pass by value versus pass by reference
      - CO5: File manipulation and IO methods
      - CO6: Problem Analysis and Design
      - CO7: UNIX systems and programming
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y					Y		
CO2	Y					Y		
CO3	Y					Y		
CO4	Y					Y		
CO5	Y					Y		
CO6	Y					Y		
CO7	Y					Y		

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1	Y							
CO2	Y							
CO3	Y							
CO4	Y							
CO5	Y							
CO6	Y							
CO7	Y							

7. Major Topics Covered in the Course:
    a. Brief history of C and of Structured Programming Paradigm.
    b. Fundamentals of modular software design.
    c. Anatomy of a C Program. Function prototypes and function definitions. Structures and Unions.
    d. Introducing pointers. Dynamic variables and memory management. Using malloc() and calloc() functions.
    e. Passing parameters by value and by reference.
    f. Dynamic arrays and pointer arithmetic.
    g. Representing Advanced Dynamic Data Structures in C
        i. Linked Lists
        ii. Stacks and Queues
        iii. Trees
    h. How to code functions for searching, sorting linked list, deleting and inserting nodes.
    i. Programming with input and output files in C.
    j. UNIX Shell Programming
    k. Process generations in UNIX, managing inter-process Communications
    l. Essential principles of Systems Programming.

1. Course number and name: CSC 2600: Computer Organization
2. Credits and contact hours: 4 credits / 82.5 hours.
3. Instructor's or course coordinator's name: Kevin Austin
4. Text book, title, author, and year: Harris and Harris, Digital Design and Computer Architecture, 2nd Edition (2013)
    a. other supplemental materials: Class notes, lab handouts, BASYS 3 reference material, SystemVerilog reference material, Logisim software, EDAplayground, Vivado
5. Specific course information
    a. brief description of the content of the course (catalog description)
    The purpose of this course is to provide a thorough discussion of the fundamentals of computer organization and architecture, and to relate these to contemporary computer design issues. After a brief review of the basic digital components used, the steps that a designer would go through in the design of an elementary system are covered followed by a discussion of the organization and design of the central processing unit (CPU) and various control system implementations. The input/output and memory subsystems are included as is a brief discussion of multiprocessing systems, pipelining, and virtual memory. Students registering for this course must also register for the accompanying Lab course.
    b. prerequisites or co-requisites: CSC1650
    c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☒required; ☐elective; ☐selected elective.

6. Specific goals for the course:
    a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic. The purpose of this course is to develop students' understanding of how digital logic elements can be combined to create a programmable computational device. It assumes that students have a knowledge of fundamental electronics, digital logic principles, and laboratory equipment. The course establishes both the theoretical and practical foundation necessary to prepare students to learn assembly language and understand microprocessor I/O operations in a subsequent course. Upon completion of the course, a student should be able to do the following:

- CO1: Demonstrate the ability to work with digital representations and understand logic circuits that implement data transfer and computation;
- CO2: Develop an arithmetic logic unit (ALU) realization;
- CO3: Simulate and implement a simple hardwired/microprogrammed controller;
- CO4: Work with a hardware definition language (HDL) to program FPGAs;
- CO5: Describe instruction set architecture, processing, opcodes and pipelining;
- CO6: Understand memory operations, parallelism, caches and virtual memory;
- CO7: Be able explain the differences between RISC, CISC, Harvard and Princeton architectures.
- CO8: Write reports to describe lab experiences.

    b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other

outcomes are addressed by the course.

	CS Program Student Outcomes							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CO1	Y						Y	Y
CO2		Y						Y
CO3		Y						Y
CO4							Y	Y
CO5							Y	Y
CO6								Y
CO7							Y	Y
CO8			Y					

7. Brief list of topics to be covered:

- Historical development
  - Generations
  - Computer level hierarchy
  - Von Neumann model
- Data representation
  - Binary, hexadecimal, codes, signed decimals
- Digital logic
  - Review of logic gates, combinational functions, sequential circuits
  - Introduction to buses, memory, clock cycles, Mealy/Moore FSMs
- Computer organization and design
- CPU Architecture, registers, datapaths
- Instruction set architecture
  - Fetch-decode cycle
  - Instruction formats, addressing modes
  - Machine language / Assembly language / Microcode
  - RISC vs. CISC
- Control unit
  - FSMs
  - Register transfer language
  - Hardwired vs. Microcoded design
- Memory
  - Types of RAM
  - Cache
  - Virtual Memory
- CPU Architectures
  - I/O bus vs. memory-mapped I/O
  - Harvard vs. Princeton architectures
  - RISC vs. CISC architectures

1. Course number and name
   CSC 3004 Parallel Programming with CUDA
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name: Brady Chen

4. Text book, title, author, and year
   CUDA by Example: An Introduction to General-Purpose GPU Programming, 1/E by Sanders & Kandrot. ISBN-10: 0131387685 ISBN-13: 9780131387683
   a. other supplemental materials
      i. CUDA Toolkit Documentation http://docs.nvidia.com/cuda/index.html#axzz3DcnSNeDt
      ii. CUDA Getting Started Guide http://docs.nvidia.com/cuda/cuda-getting-started-guide-for-microsoft-windows/index.html#axzz3DcnSNeDt
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This topics course covers programming techniques on the parallel computing architecture and programming model, CUDA. The topics include the introduction of CUDA, the basic CUDA commands and syntax, as well as several optimizations for CUDA code and utilization of CUDA libraries.
   b. prerequisites or co-requisites
      CSC2560 Systems Programming
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Elective
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      - CO1: Basic concepts of parallel processing and CUDA architecture.
      - CO2. CUDA C language.
      - CO3. Parallel programming in CUDA C
      - CO4. Thread cooperation including thread communications and synchronizations.
      - CO5. Memory usage and management including constant memory and texture memory.
      - CO6. Graphics interoperability
      - CO7. Performing arithmetic with atomic operations in your CUDA C kernels
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	X	X				X		
CO2	X	X				X		
CO3	X	X				X		
CO4	X	X				X		
CO5	X	X				X		
CO6	X	X				X		
CO7	X	X				X		

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1	X	X						
CO2	X	X						
CO3	X	X						
CO4	X	X						
CO5	X	X						
CO6	X	X						
CO7	X	X						

7. Brief list of topics to be covered
   - Introduction to CUDA architecture
   - GPU hardware and parallel communication patterns
   - Parallel Programming in CUDA C
   - Thread Cooperation
   - Constant Memory and Events
   - Texture Memory
   - Graphics Interoperability
   - Atomics
   - Streams
   - CUDA C on Multiple GPUs

1. Course number and name: CSC 3011 Data Modeling and Database Design
2. Credits and contact hours: 3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name: Ricky J. Sethi
4. Text book, title, author, and year
   Modern Database Management, Tenth Edition, Hoffer /Ramesh/Topi. ISBN: 9780136088394.
   a. other supplemental materials
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course is about design and implementation of database systems. Evolution of various database models including hierarchical, relational and object-oriented models and the advantages of different models are studied. Use of Structured Query Language (SQL) in relational databases is explained and applied. Students design databases applying E-R modeling and normalization techniques.
   b. prerequisites or co-requisites
      CSC 1550 and CSC/MATH 1900
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1. The evolution of various database models
      CO2. Understand various database operators
      CO3. Advanced SQL commands
      CO4. E-R modeling
      CO5. Normalization
      explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1						Y		
CO2	Y					Y		
CO3	Y					Y		
CO4	Y					Y		
CO5						Y		

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1						Y		
CO2						Y		
CO3						Y		
CO4						Y		
CO5						Y		

7. Brief list of topics to be covered
    a. Introduction to Databases
        i. File systems
        ii. Database systems
        iii. Database models
    b. Relational Database Model
        i. Entities and attributes
        ii. Relational database keys and indexes
        iii. Integrity rules
        iv. Data dictionaries
    c. Structures Query Language
    d. E-R Modeling
    e. Normalization
    f. Database Life Cycle
        i. Information system overview
        ii. System development life cycle
        iii. Planning, analysis and design of systems
        iv. Implementation and maintenance
        v. Database design strategies
        vi. Security, backup and recovery and concurrency control.
    g. Object-Oriented Database Management (5 hrs)
        i. Object-oriented design concepts
        ii. Object schemas
        iii. Representing relationships
        iv. Features of an object-oriented database management systems (OODBMS)
    h. Client/Server Systems (2.5 hrs)
        i. Need for client/server systems
        ii. 3-tier architecture of client/server systems

1. Course number and name: CSC 3012 Introduction to Data Science
2. Credits and contact hours: 3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name: Ricky J. Sethi
4. Text book, title, author, and year
   Data Science from Scratch By Joel Grus 1st Edition by O'Reilly; ISBN : 149190142X
   a. other supplemental materials
5. Specific course information
   a. brief description of the content of the course (catalog description)
      Human beings now generate, and collect, massive amounts of data. As big data becomes the norm in modern business and research, there is a growing demand for individuals who are able to make decisions and derive meaningful insight from large-scale, heterogeneous data. A data scientist is a person who has the skills, knowledge, and ability to extract actionable knowledge from the raw data. This course will cover the topics needed to solve data-science problems, which include data preparation (collection & integration), data characterization and presentation, and data analysis (experimentation & observational).
   b. prerequisites or co-requisites
      CSC 1550 and CSC 2560
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Elective
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1. What is Data Science? Identifying questions and developing an empirical framework
      CO2. Basics of a programming language like Python or R
      CO3. Exploring Data, including Data capture, storage, and organization
      CO4. Basic Statistics, Review of Probability for traditional analysis, and Distributions, tests, and the importance of basic statistics
      CO5. Inferential Statistics: Learning from data, including Bootstrapping and resampling and Regression
      CO6. Advanced analytics: Cross-validation and also Data errors: Correlation, overfitting, and multiple comparisons
      CO7. Machine Learning Introduction, including Supervised Learning and Unsupervised Learning

      explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y							
CO2	Y							
CO3	Y							
CO4	Y							
CO5	Y					Y		
CO6	Y							
CO7	Y					Y		

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1		Y		Y		Y		
CO2	Y							
CO3						Y		
CO4	Y							
CO5	Y							
CO6	Y	Y						
CO7	Y	Y						

7. Brief list of topics to be covered
    a. What is Data Science?
    b. A Crash Course in Python
    c. Exploring Data
    d. Identifying questions and developing an empirical framework
    e. Data capture, storage, and organization
    f. Basic analytics and Review of Probability for traditional analysis
    g. Inferential Statistics: Learning from data
    h. Distributions, tests, and the importance of basic statistics
    i. Bootstrapping and resampling
    j. Regression
    k. Advanced analytics: Cross-validation
    l. Data errors: Correlation, overfitting, and multiple comparisons
    m. Machine Learning Introduction
    n. Supervised Learning
    o. Unsupervised Learning

1. Course number and name
   CSC 3040 Cybersecurity Management
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name: Audrey Pereira

4. Text book, title, author, and year
   Corporate Computer Security By Boyle, R..J, & Panko, R..R. 4th Edition, ISBN : 9780133545197
   a. other supplemental materials

5. Specific course information
   a. brief description of the content of the course (catalog description)
      This is an introductory course in information technology (IT) security that incorporates managerial, criminal justice, and IT components. This course focuses on the management and legal issues and practical implications related to securing corporate information systems. Students will explore areas including IT security threats, security management processes, compliance laws and regulations, risk analysis and how to respond to risk, network security, WWW and e-commerce security, and incident and disaster recovery. Cross-listed as BSAD 3040 and CJ 3040.
   b. prerequisites or co-requisites
      BSAD 1700 or CSC 1400
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Elective
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The specific outcomes of instruction are address by the following course objectives:
      - CO1: Explain key security concepts related to IT and use basic security terminology and acronyms correctly
      - CO2: Describe threats from multiple areas, including employees, ex-employees, malware writers, and other criminals
      - CO3: Explain the plan-protect-respond security management cycle
      - CO4: Understand basic networking and cryptography concepts, and the importance of securing networks
      - CO5: Explain the basics of disaster response, including business continuity planning.
      - CO6: Make intelligent, reasonable, thoughtful, and accurate decisions about IT security, vulnerabilities, and legal issues, including criminal matters.
      - CO7: Keep current on security related issues by describing and discussing recent newsworthy security events.
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1				X				
CO2				X				
CO3	X							
CO4	X	X						
CO5	X	X						
CO6	X	X		X				
CO7	X			X				

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1				X		X		
CO2				X				
CO3	X							
CO4	X	X				X		
CO5	X	X				X		
CO6	X	X		X				
CO7	X			X				

7. Brief list of topics to be covered
   - Planning and Policy - Compliance Laws and Regulations, Organization, Risk Analysis, Technical Security Architecture, Policy-Driven Implementation, Governance Frameworks
   - Chapter 3 Cryptography - What is Cryptography? Symmetric Key Encryption Ciphers, Cryptographic System Standards, The Negotiation Stage, Initial Authentication Stage, The Keying Stage, Message-By-Message Authentication
   - Secure Networks - DoS Attacks, ARP Poisoning, Access Control for Networks, Ethernet Security, Wireless Security
   - Access Control - Physical Access and Security, Passwords, Access Cards and Tokens, Biometric Authentication, Cryptographic Authentication, Authorization, Auditing, Central Authentication Servers, Directory Servers, Full Identity Management
   - Chapter 6 Firewalls 313 - Static Packet Filtering, Stateful Packet Inspection, Network Address Translation, Application Proxy Firewalls and Content Filtering, Intrusion Detection Systems and Intrusion Prevention Systems, Antivirus Filtering and Unified Threat Management, Firewall Architectures, Firewall Management
   - Application Security - WWW and E-Commerce Security, Web Browser Attacks, E-Mail Security, Voice over IP Security, Other User Applications
   - Incident and Disaster Response - The Intrusion Response Process for Major Incidents, Intrusion Detection Systems, Business Continuity Planning, It Disaster Recovery

1. Course number and name
   CSC 3050 Web Programming
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Nadimpalli Mahadev
4. Text book, title, author, and year
   No textbook.
   a. other supplemental materials
      zyBooks for Web Programming and Web Programming Notes.
5. Specific course information
   a. brief description of the content of the course (catalog description)
      There are three tiers to Web Programming: Front-end GUI programming, back-end database design with business logic programming and a middle-tier web server with control logic programming. In this course, students complete a web project creating this 3-tier architecture by learning the latest programming languages used in web development.
   b. prerequisites or co-requisites
      CSC 1550
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Elective
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1: Types of Web Applications.
      CO2: The role of Hypertext Markup Language and platform independence.
      CO3: Use of Cascading Style Sheets in HTML.
      CO4: JavaScript and form validation.
      CO5: Three-tier architecture and MVC architecture.
      CO6: Coding with Java Server Pages.
      CO7: Database design and the role of JDBC.
      CO8: Creating JavaBeans for business logic.
      CO9: ER diagrams and entity class diagrams
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes:

	Computer Systems Student outcomes (CSSO)						
	1	3	4	5	6	7	8
CO1					x		
CO2	x				x		
CO3	x						
CO4	x						
CO5						x	
CO6	x						
CO7					x	x	
CO8	x					x	
CO9					x	x	

Mapping of course objectives to CIS student outcomes:

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1								
CO2								
CO3							x	
CO4							x	
CO5		x					x	
CO6								
CO7		x				x	x	
CO8						x	x	
CO9		x				x	x	

7. Brief list of topics to be covered
    a. Overview of Java Programming Language.
    b. Overview of web programming tools: HTML, CSS, JavaScript, CSS and JavaScript Libraries and Java Server pages.
    c. Client-side scripting with JavaScript.
    d. NetBeans configuration for web development.
    e. Overview of Java Server Pages and form processing with JSP.
    f. Scripting with JSP elements.
    g. Creating and using Java Beans with JSP.
    h. Relational database design.
    i. Database programming using SQL, JDBC and JSP.

1. Course number and name
   CSC 3100 Operating Systems
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Brady Chen
4. Text book, title, author, and year
   Modern Operating Systems 4th Edition by Andrew S. Tanenbaum and Herbert Bos, 2015
   a. other supplemental materials
      none
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course examines the internal structure and operation of operating systems with an emphasis on their design criteria and approaches. Topics covered include: process management, scheduling, deadlock, memory management, virtual memory, protection and security and distributed systems. A working knowledge of a higher-level system programming language and computer data structures is assumed.
   b. prerequisites or co-requisites
      CSC2560 Systems Programming
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The specific outcomes of instruction are address by the following course objectives:
      CO1: Learn basic principles and components of an Operating System.
      CO2. Learn and be familiar with the notion of processes.
      CO3. Understand CPU scheduling, deadlock detection and deadlock avoidance.
      CO4. Understand how memory management is handled in an operating system.
      CO5. Understand how file systems are implemented.
      CO6. Learn and be familiar with the operating system security issues.
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	X							
CO2	X							
CO3	X							
CO4	X							
CO5	X							
CO6	X							

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1	X							
CO2	X							
CO3	X							
CO4	X							
CO5	X							
CO6	X							

7. Brief list of topics to be covered
    a. Operating System concepts OS1(2.5) OS2(2.5)
        i. Computer hardware review
        ii. Operating system concepts and structure
    b. Processes and threads OS3(6) OS4(4)
        i. Processes and threads
        ii. Inter-process communication (IPC) and classic IPC problems
        iii. CPU scheduling
    c. Deadlocks OS3(4)
        i. Introduction to deadlock
        ii. Deadlock detection and recovery
        iii. Deadlock avoidance
        iv. Deadlock prevention
    d. Memory management OS5(6)
        i. Basic memory management
        ii. Virtual memory
        iii. Page replacement algorithm
    e. Input/output OS6(4)
        i. I/O hardware
        ii. I/O software
    f. File systems OS8(4)
        i. Files
        ii. Directories
    g. Operating system security OS7(4)

1. Course number and name: CSC 3200: Programming Languages
2. Credits and contact hours: 3 credits / 37.5 hours.
3. Instructor's or course coordinator's name: Natasha Kurtonina
4. Text book, title, author, and year:
   Essentials of Programming Languages, Daniel P. Friedman and Mitchell Wand ISBN-13: 978-0262062794 ISBN-10: 0262062798
   a. other supplemental materials:
      Notes, handouts and lectures are posted on Blackboard for student review.

5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course provides an introduction to modern study of programming languages and compiler construction. It explores the logic of languages construction, there places in programming paradigms along with their theoretical analysis and software tools. We will explore all major programming paradigms, their philosophies and technical implementations. Various programming techniques, algorithms and data structures will be studies and compared in different from different programming frameworks perspectives.
   b. prerequisites or co-requisites:
      CSC 2560 (Systems Programming)
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☒required; ☐elective; ☐selected elective.

6. Specific goals for the course:
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic. The purpose of this course is to give students a foundation in programming languages of main programming paradigms. The course covers the theory behind programming paradigms, introduces the most important programming techniques useful for modern software design and provides in depth analysis of commonly used programming languages. Students will learn underlying mathematical foundation of programming languages and programming paradigms. Upon completion of the course, a student should be able to understand the theoretical foundations and programming implementations of
      - CO1: Functional, logical & procedural paradigms
      - CO2: Programming paradigms strengths and weaknesses
      - CO3: Implementation structures for the paradigms
      - CO4: Functional programming using Scheme
      - CO5: Logic programming using Prolog
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y					Y		
CO2	Y					Y		
CO3	Y					Y		
CO4	Y					Y		
CO5	Y					Y		

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1	Y							
CO2	Y							
CO3	Y							
CO4	Y							
CO5	Y							

7. Major Topics Covered in the Course:
    a. Introduction to Programming Paradigms
    b. Logic Programming, Prolog and AI
    c. Functional Programming: examples and foundations
    d. Recursion and Inductive Sets of Data Logic and Functional Programming with Scheme
    e. Data Abstraction
        i. Specifying Data via Interfaces
        ii. Representation Strategies for Data Types
        iii. Interfaces for Recursive Data Types
        iv. A Tool for Defining Recursive Data Types
        v. Abstract Syntax and Its Representation Advanced Design
    f. Grammars and Parsing Techniques.
    g. Comparative analysis of C++, Java and C#. Objects and Classes
        i. Object-Oriented Programming: principles and implementations
        ii. Inheritance and Interface
        iii. Compilers and Interpreters for OOP
        iv. Types and Types Checkers
        v. Aspect Oriented Programming
    h. Major Data Structures in Various Programming Paradigms
    i. Major Algorithmic techniques in Various Programming Paradigms
    j. Programming Languages and Web design

k. Programming Languages and Artificial Intelligence

1. Course number and name: CSC 3300: Assembly Language Programming
2. Credits and contact hours: 3 credits / 37.5 hours.
3. Instructor's or course coordinator's name: Kevin Austin
4. Text book, title, author, and year Hyde, Randall, The Art of Assembly Language:
    a. other supplemental materials: Various topic handouts and practice problems, in-class notes and screen-capture video of lectures are posted on Blackboard for student review.
5. Specific course information
    a. brief description of the content of the course (catalog description)
    Assembly language provides the means for programming a computer at the most basic machine level. In this course, we explore the fundamental operations of a modern computer system using software tools. Topics examined include numerical and character representations, microprocessor register usage, machine instructions, addressing modes, input/output processing, parameter passing, interrupt processing and simple data structure realizations on the Intel 80x86 processor.
    b. prerequisites or co-requisites: CSC1550
    c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☐required; ☒elective; ☐selected elective.
6. Specific goals for the course:
    a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic. The purpose of this course is to develop students' understanding of digital computer organization and gain an appreciation for high-level language constructs through assembly language programming. Having successfully completed this course, a student will have:

- CO1: demonstrated how information is represented and operated upon in a digital computer;
- CO2: demonstrated knowledge of how information is processed at the machine level within a CPU;
- CO3: analyzed assembly language code sequences to perform arithmetic, calculate memory addresses, determine processor flags, predict conditional branches;
- CO4: written assembly language programs to perform console I/O;
- CO5: written assembly language programs that draw directly to screen memory;
- CO6: written assembly language programs that demonstrate bit masking techniques;
- CO7: written assembly language programs that work with arrays using indexed addressing;

    b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

	CS / CIS Program Student Outcomes							
	1	2	3	4	5	6	7	8
CO1							Y	
CO2							Y	
CO3								
CO4		Y						
CO5		Y						
CO6		Y						
CO7		Y						

7. Brief list of topics to be covered:

- Digital representations
    - Bits, bytes, nibbles, words
    - Binary, hexadecimal, BCD, ASCII and two's complement representations
- Architecture
    - Components: CPU, data paths, memory, registers, ALU, I/O devices
    - Processes: Addressing, instruction fetch/decode
    - Programmer's register model, real mode memory map
- Real mode registers / hardware stack:
    - General purpose, segment, stack pointer, program counter, index
    - Status, processor flags, stack operations
- Assembly Language Fundamentals
    - Program structure, entry and exit
    - Variable and constant declaration directives
    - Fundamental instructions: MOV, ADD, SUB, NEG, AND, OR, NOT, XOR
    - System software interrupt calls
- Subroutines and software interrupts
- Context saving, instruction results and processor flags
- Branching: conditional / unconditional + loops, relative / absolute addressing
- Addressing modes and data structures: Immediate, direct, indirect, based, indexed
- Arrays / structs
- I/O devices: Keyboard, display, file I/O

1. Course number and name: CSC3350: Small-scale Embedded Systems Development
2. Credits and contact hours: 3 credits / 37.5 hours.
3. Instructor's or course coordinator's name: Kevin Austin
4. Text book, title, author, and year: Embedded C Programming: Techniques and Applications of C and PIC MCUS by Mark Siegesmund (Newnes © 2014 ISBN: 9780128013144). Other PICC reference material available from ACM Learning Center online.
    a. other supplemental materials: Various topic handouts, device data sheets, detailed lab exercises. In-class notes and screen-capture video of lectures are posted on Blackboard for student review. Hardware: LabX-1 PIC microcontroller platforms, PicKit device programmers, RS-232 to USB converters.
5. Specific course information
    a. brief description of the content of the course (catalog description)
    Small-scale embedded systems are everywhere. Students will learn about how to develop functional devices using microcontrollers and become familiar with microcontroller architecture and the various function-specific hardware modules such as timer/counters, analog-to-digital converters, pulse-width modulators, liquid-crystal displays and keypads. Hands-on projects will require the writing of device drivers and the development of a useful embedded system.
    b. prerequisites or co-requisites: CSC1650 and CSC2560
    c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☐required; ☒elective; ☐selected elective.

6. Specific goals for the course:
    a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.

The purpose of this course is to develop students' understanding of structured software design and iterative development methods for embedded applications. Students work in pairs to develop device drivers in the C programming language for microcontroller-based applications. Having successfully completed this course, a student will have:

- CO1: demonstrated understanding of the top-down problem analysis and structured device driver design process;
- CO2: measured interrupt overhead and demonstrated an understanding of interrupt processing;
- CO3: analyzed and programmed several hardware-based timer configurations;
- CO4: demonstrated an understanding of how to improve code readability using C language structs to create variables that represent bit-fields within register;
- CO5: written, tested and documented a device driver to operate an LCD display;
- CO6: written, tested and documented a device driver to scan a keypad matrix;
- CO7: written, tested and documented a device driver to implement a synchronous serial interface to a memory device;

    b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

	CS Program Student Outcomes							
	1	2	3	4	5	6	7	8
CO1					Y		Y	
CO2					Y			
CO3					Y	Y		
CO4					Y		Y	
CO5		Y	Y		Y	Y		
CO6		Y	Y		Y	Y		
CO7		Y	Y		Y	Y		

7. Brief list of topics to be covered:
   - Embedded Software Basics
     - PICC int data types / bit-field structs
     - Mapping variables to memory
     - Transferring code to the CPU
   - Embedded application structure
   - PIC Microcontroller Architecture
     - CPU, Busses, RAM and ROM
     - Instruction pipeline
     - I/O ports, embedded devices
     - Harvard / RISC architecture
     - Clock generation
   - PIC assembly language overview / comparison with C
   - PICC Interrupt processing:
     - service routine responsibility, overhead
     - limited hardware stack
     - foreground / background task communication
   - Hardware timers
     - Tone generation, Interrupt pacing
     - Switch de-bouncing
     - Pulse-width modulation
   - External Device Interfaces
     - Parallel
     - Asynchronous/Synchronous serial
   - Structured Software Design / Incremental development
     - Requirements definition
     - Top-down design/Bottom-up development
     - Testing / debugging methods
   - Device Driver Projects
     - Loudspeaker
     - LCD display
     - Keypad matrix scanning
     - Serial EEPROM

1. Course number and name
   CSC 3400 Data Communications & Networking
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Brady Chen
4. Text book, title, author, and year

   Communications and Networking: An Introduction, 2nd Edition by John Cowley. Springer Publishing Company, Incorporated ©2012
   ISBN:1447143566 9781447143567
   a. other supplemental materials
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course investigates the means by which data is exchanged by two digital devices. Topics include the history of data communications, the public switching network (PSTN), standards bodies (OSI, IEEE, etc), serial synchronous/ asynchronous data flow, channel characteristics (bandwidth, noise, capacity, physical implementations), modulation techniques (modems and standards), circuit and packet switching (Asynchronous Transfer Mode (ATM)), multiplexers, Integrated Service Digital Network (ISDN), Digital Subscriber Lines (DSL), etc. An introduction to Wide Area Networks (WAN) is included.
   b. prerequisites or co-requisites
      CSC 2560
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO12. Understand and describe high level network concepts;
      CO13. Describe how agents on a network communicate;
      CO14. Understand how physical media is used;
      CO15. Understand the following about Local Area Networks; (7)
      CO16. Understand the following about Wide Area Networks; (8)
      CO17. Be familiar with TCP/IP - the internet protocol; (9)
      CO18. Be familiar with these applications from the Internet Application Layer;
      CO19. Understand the following about Network Security;
      CO20. Be familiar with the following concepts of Network Management;
      CO21. Understand and describe the following Wireless Networks
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y				Y			
CO2	Y				Y			
CO3	Y				Y			
CO4	Y				Y		Y	
CO5	Y				Y		Y	
CO6	Y				Y		Y	

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1					Y			Y
CO2					Y			Y
CO3					Y			Y
CO4					Y		Y	Y
CO5					Y		Y	Y
CO6					Y		Y	Y

7. Brief list of topics to be covered
    a. Server installation: Hardware requirements, Partitions & installation options, Installation & testing
    b. Server configuration: Control panel, Server environment, Devices & resources
    c. Storage, backup & performance
    d. Accounts & client connectivity
    e. Security: Server resources and security, Objects and security, Moving files/folders
    f. File Systems and Disk Quotas: Distributed file system, Shares, Disk quotas, Application installations, Samba, etc.
    g. LAN configuration and protocols: Media, Network Interface card, Ethernet, token ring, Bus, ring, star topology, NetBEUI, IPX, TCP/IP
    h. Server hardware planning: Compatibility, Bus speed, CPU, Disk, Memory, NIC
    i. Network protocol planning: NDIS, ODI, TCP/IP, NWLink, NetBEUI, DLC, AppleTalk
    j. Server monitoring and optimization: Logs and alerts
    k. Network monitoring and tuning

1. Course number and name: CSC 3450 Local Area Networks
2. Credits and contact hours: 3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name: Ricky J. Sethi
4. Text book, title, author, and year
   Networking Basics, 2nd Edition, ISBN: 9780619055820, Mark Ciampa
   Linux+ Guide to Linux Cert, 3rd Edition, ISBN: 9781418837211, Jason W. Eckert
   a. other supplemental materials
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course examines local area network (LAN) technology and architecture both through general concepts and practical hands-on experience. All networking fundamentals are presented based on the modular approach of the ISO standards. Topics covered include file servers, configurations and protocols, installation and management of server hardware and software, system monitoring, maintenance and troubleshooting. Due to its importance, the TCP/IP protocol will be stressed.
   b. prerequisites or co-requisites
      CSC 2560
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO8. Networking topologies and access methods;
      CO9. Network architectures and standards;
      CO10. Network protocols, hardware and software;
      CO11. Network planning;
      CO12. Server installation and configuration;
      CO13. Network management, monitoring and troubleshooting
      explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	CS Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y				Y			
CO2	Y				Y			
CO3	Y				Y			
CO4	Y				Y		Y	
CO5	Y				Y		Y	
CO6	Y				Y		Y	

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1					Y			Y
CO2					Y			Y
CO3					Y			Y
CO4					Y		Y	Y
CO5					Y		Y	Y
CO6					Y		Y	Y

7. Brief list of topics to be covered
    a. Server installation
        i. Hardware requirements
        ii. Partitions & installation options
        iii. Installation & testing
    b. Server configuration
        i. Control panel
        ii. Server environment
        iii. Devices & resources
    c. Storage, backup & performance
    d. Accounts & client connectivity
    e. Security
        i. Server resources and security
        ii. Objects and security
        iii. Moving files/folders
    f. File Systems and Disk Quotas
        i. Distributed file system
        ii. Shares
        iii. Disk quotas
        iv. Application installations
        v. Samba, etc.
    g. LAN configuration and protocols
        i. Media
        ii. Network Interface card
        iii. Ethernet, token ring
        iv. Bus, ring, star topology
        v. NetBEUI, IPX, TCP/IP
    h. Server hardware planning
        i. Compatibility, Bus speed, CPU, Disk, Memory, NIC
    i. Network protocol planning
        i. NDIS, ODI, TCP/IP, NWLink, NetBEUI, DLC, AppleTalk
    j. Server monitoring and optimization
        i. Logs and alerts
    k. Network monitoring and tuning

1. Course number and name: CSC 3600: Microprocessors
2. Credits and contact hours: 4 credits / 82.5 hours.
3. Instructor's or course coordinator's name: Kevin Austin
4. Text book, title, author, and year: Numerous reference works are used. Students use the ACM Learning Center to access materials online. Varies from semester to semester.
    a. other supplemental materials: PIC microcontroller data sheets, online Arduino UNO reference material, course-specific topic notes, lab handouts.
5. Specific course information
    a. brief description of the content of the course (catalog description)
    This course provides a detailed study of the microprocessor and its applications. Emphasis is placed on a current microprocessor, its hardware and software and its associated family of integrated circuits. Students design a microprocessor system, configuring the random access memory, the read-only memory, and peripheral devices using peripheral interface adapters. Students reinforce theory with extensive laboratory work. Students registering for this course must also register for the accompanying lab course.
    b. prerequisites or co-requisites: CSC 2600
    c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☒required; ☐elective; ☐selected elective.

6. Specific goals for the course:
    a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.

The purpose of this course is to develop students' understanding of microprocessor-based systems. Students develop an understanding of hardware/software interaction through numerous laboratory experiences. This course introduces students to various communication protocols, programming paradigms and tradeoffs associated with programming both the microprocessor and the I/O devices themselves. Students write programs in both assembly language and C to acquire data and operate a variety of modern I/O devices. Upon successful completion of this course, a student will have demonstrated knowledge of:

- CO1: Machine architecture, operations and hardware stack implementation
- CO2: Assembly language programming
- CO3: Electrical characteristics of CPU and I/O devices
- CO4: I/O programming data structures and techniques
- CO5: I/O device data transfer and processing protocols
- CO6: Hardware interrupt processing
- CO7: I/O device communication protocols
- CO8: Working with a partner toward a common goal and writing reports.

    b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

	CS Program Student Outcomes							
	1	2	3	4	5	6	7	8
CO1								Y
CO2							Y	Y
CO3		Y						Y
CO4	Y	Y				Y		Y
CO5	Y	Y						Y
CO6		Y						Y
CO7	Y	Y				Y	Y	Y
CO8			Y		Y			

7. Brief list of topics to be covered:
   - Microprocessor overview
     - CPU's, busses, memory, clock, ports
     - Number systems and data formats
   - PIC microcontroller architecture
     - Program memory, register file
     - Key registers: configuration, program counter, status, w
     - Clock, I/O port configuration, internal peripherals, shadowing
   - Assembly language
     - Program structure: mnemonics / opcodes / directives
     - Instruction types: transfer, compute, branch
     - Stack operations: subroutines & interrupts
   - Hardware specifications
     - Port pin electrical specifications and configuration
     - Machine cycles, instruction pipeline
     - Memory: program memory busses, register file busses
     - Interrupt processing: latency, overhead, efficiency
   - Basic IO port interfacing
     - I/O pins: configuration, bit masking
     - Programming input pins / handling switch bounce
     - Basic Interrupt handling / Vectors for interrupt and device reset
     - Interrupt control / ISR responsibilities and best practices
   - Standard communication protocols
     - Parallel interfaces
     - Asynchronous serial interfaces (UART)
     - Synchronous serial interfaces (SPI, I2C)
   - I/O programming techniques
     - Data structures / bit masking
     - Interrupt-driven I/O vs. I/O polling vs. DMA
     - Data conversion: sampling and resolution
   - Devices
     - Pushbutton and LED
     - LCD display, Bluetooth communication

- Temperature & inertial measurement

1. Course number and name: CSC 3700: Algorithms and Data Structures
2. Credits and contact hours: 3 credits / 37.5 hours.
3. Instructor's or course coordinator's name: Natasha Kurtonina
4. Text book, title, author, and year:
   Introduction to Algorithms. T.H. Cormen, C.E. Leiserson R.I. Rivest. MIT Press
   ISBN: 9780262033848
   a. other supplemental materials:
      Notes, handouts and lectures are posted on Blackboard for student review.

5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course provides an introduction to modern study of computer algorithms and complex data structures. It explores the logic of algorithmic analysis and design as well as practical utility and implementation techniques. The basic ideas from Complexity Theory will help students understand the concept of efficiency of algorithms and its role in the design of algorithms. The analysis of data structures is focused on the concepts of linked list, queues, stacks and trees. Various programming techniques such as divide and conquer; dynamic programming and backtracking are presented.
   b. prerequisites or co-requisites:
      CSC 2560 (Systems Programming)
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☒required; ☐elective; ☐selected elective.

6. Specific goals for the course:
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The purpose of this course is to give students a foundation in design and analysis of the most important algorithms and data structures. Upon successful completion of this course, a student will have demonstrated knowledge of:
      - CO1: Sorting and Search Algorithms
      - CO2: Graph Theory Algorithms
      - CO3: Cryptography Algorithms
      - CO4: Dynamic Data Structures and related algorithms
      - CO5: Fundamentals of Computational Complexity Theory
      - CO6: Algorithmic Design Paradigms

   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	Computer Science Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1	Y					Y	Y	
CO2	Y					Y	Y	
CO3	Y					Y	Y	
CO4	Y					Y	Y	
CO5	Y					Y	Y	
CO6	Y					Y	Y	

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1	Y						Y	
CO2	Y						Y	
CO3	Y						Y	
CO4	Y						Y	
CO5	Y						Y	
CO6	Y						Y	

7. Major Topics Covered in the Course:
    a. Introduction to Complexity Analysis of Algorithms
    b. Fundamental Searching and Sorting Algorithms: analysis and principles of implementation. Complexity of searching and sorting algorithms. Implementations in various programming paradigms.
    c. Data Structures and Related Algorithms. Linked Lists, Stacks and Queues, Binary Search Trees, Red-Black Trees Spanning Trees, Hash Tables.
    d. Advanced Design and Analysis Techniques
    e. Dynamic Programming
    f. Greedy Algorithms
    g. Backtracking Techniques
    h. Graph Algorithms
    i. Cryptographic Algorithms
    j. Computability and NP-Completeness

1. Course number and name: CSC4210: Computer Graphics Programming
2. Credits and contact hours: 3 credits / 37.5 hours.
3. Instructor's or course coordinator's name: Kevin Austin
4. Text book, title, author, and year: Computer Graphics Programming in OpenGL With C++, by V. Scott Gordon and John Clevenger (2019)
    a. other supplemental materials: Students need Visual Studio 2017 installed on their personal laptop computers. We also use the tutorials at these web sites: http://www.opengl-tutorial.org and https://learnopengl.com/ .
5. Specific course information
    a. brief description of the content of the course (catalog description)
    In this course, we study the processes, tools and mathematics that underlie the production of images on digital computers. Students will write programs that create digital representations of geometric objects and render them on a computer screen. Topics covered in this course are: the graphics processing pipeline, graphics primitives, vertex specification, rendering, color theory, geometric transformations, illumination, perspective and the virtual camera. In the process, students will gain practical experience with programming concepts related to abstraction, data structures, algorithms and memory management.

    b. prerequisites or co-requisites: CSC2560
    c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☐required; ☒elective; ☐selected elective.

6. Specific goals for the course:
    a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.

In this course, students learn the fundamentals of rendering 3-D object models on a computer screen using OpenGL. After having completed this course, students will have demonstrated:

CO1: An appreciation for the history and evolution of computer graphics.
CO2: Programs that implement a graphics pipeline to render images on a computer screen.
CO3: Created hard-coded models of simple geometric shapes and applied textures to them.
CO4: Programs that implement various geometric transformations of scale, translation, and rotation.
CO5: An understanding of how model, view and projection matrices can be created and manipulated to render images in a variety of ways.
CO6: Programs that implement and manipulate scene lighting and reflective properties of a model.
CO7: Programs that implement transparency and shadows.

    b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

	CS / CIS Program Student Outcomes							
	1	2	3	4	5	6	7	8
CO1							Y	
CO2						Y	Y	
CO3							Y	
CO4						Y		
CO5							Y	
CO6						Y		
CO7						Y		

7. Brief list of topics to be covered:

- Evolution of computer graphics hardware
- The graphics pipeline and the role of shader programs
- The OpenGL API and support libraries: GLEW, GLFW and GLM
- Colors, color blending, transparency
- 3-D representations, coordinate systems, vertices
- Coordinate systems: vertices and models
- Triangles, matrices and vectors
- Data structures: Vertex Array Objects, Vertex Buffer Objects
- Rotation, translation and scaling
- Point-of-view: camera location, perspective, frustum
- MVP matrix, uniforms, GLSL, the depth buffer
- Applying textures, UV coordinates
- Normal vectors
- Applying illumination, reflection and shadows

1. Course number and name
   CSC 3710 Systems Analysis Methods
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Audrey Pereira & Brady Chen
4. Text book, title, author, and year
   Systems Analysis and Design (9th ed.). by Kendall, K. E., & Kendall, J. E., 2014
   a. other supplemental materials

5. Specific course information
   a. brief description of the content of the course (catalog description)
      As an overview of the system development life cycle, this course introduces the student to the fundamental concepts and techniques of systems analysis. Classical and structured methods of systems documentation are explored as well as information gathering and reporting activities.
   b. prerequisites or co-requisites
      CSC3450 Local Area Networks
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The specific outcomes of instruction are address by the following course objectives:
      CO1: Describe the systems development life cycle and specific life cycle models
      CO2: Describe systems analysis and the role of the systems analyst
      CO3: Describe how information systems projects are proposed and initiate
      CO4: Develop basic systems documentation including project charters, system proposals, requirements questionnaires, prototypes, event response tables, and context level diagrams
      CO5: Analyze, model, and specify a system's process and data requirements
      CO6: Compare and contrast structured and object oriented development
      CO7: Discuss emerging trends and issues in systems analysis
      CO8: Work cooperatively in a group to integrate the concepts learned
      CO9: Construct and present effective oral and written forms of professional communications.
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1		X						
CO2		X						
CO3		X						
CO4		X						
CO5		X						
CO6	X							
CO7							X	
CO8					X			
CO9							X	

7. Brief list of topics to be covered
    a. Systems development life cycle and specific life cycle models
        i. Systems development life cycle (problem identification, determining information requirements, analyzing system needs, designing systems, developing and documenting software, testing and maintaining system, and implementing and evaluating system)
        ii. Agile approach and objective-oriented systems analysis and design
        iii. Choosing which systems development method to use
    b. Systems analysis and the role of the systems analyst, and the analysis process
        i. Need for analysis and design
        ii. Role of a systems analyst
        iii. Developing data flow diagrams
        iv. Logical and physical data flow diagrams
    c. How information systems projects are proposed and initiated and project management
        i. Project initiation, determining feasibility
        ii. Hardware and software needs
        iii. Identifying, forecasting, and comparing costs and benefits
        iv. Project schedule and managing time, activities, and project team
        v. Controlling and managing a project
    d. Development of basic systems documentation
        i. Depicting systems graphically
        ii. Systems proposal, use case development, questionnaires
        iii. Data flow diagrams (logical and physical), data dictionaries
    e. Process and data requirements, modeling systems (information requirements analysis)
        i. Organizations as systems
        ii. Use case modeling
        iii. Interviewing, listening to stories, and questionnaires
        iv. Joint application design, process specifications
        v. Structured English, Decision tables, and decision trees
    f. Structured and object-oriented development
        i. Agile modeling
        ii. Object-oriented systems analysis and design
        iii. Prototyping
        iv. Comparing Agile modeling and structured methods
    g. Emerging trends and issues in systems analysis
        i. Current events discussed as they emerge

1. Course number and name
   CSC 4005 Ethical Hacking
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Nadimpalli Mahadev
4. Text book, title, author, and year

   Hacker Techniques, Tools, and Incident Handling, 3rd edition by S-P Oriyano and M.G. Solomon, Jones & Bartlett Learning 2020.
   a. other supplemental materials
      Class Notes and other handouts posted to Blackboard.
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course introduces the students to working with hacking tools and strategies in order to detect and report the vulnerabilities in computer systems such as networks and operating systems. Topics include the four stages of penetration testing: reconnaissance, scanning, exploitation and maintenance. Students will generate reports at each stage. Necessary background in networking and defensive programming will be reviewed.
   b. prerequisites or co-requisites
      CSC 2560 and CSC 3450
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Elective
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1: Dangers of unethical hacking.
      CO2: Network security fundamentals.
      CO3: OSI and TCP/IP models.
      CO4: Various stages of ethical hacking.
      CO5: Passive Reconnaissance tools and techniques.
      CO6: Port scanning and network mapping tools and techniques.
      CO7: Vulnerability detection tools and exploring known vulnerabilities lists (CVE).
      CO8: Vulnerability exploitation tools.
      CO9: Tools for maintaining access.

   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes:

	CS Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1				x				
CO2							x	
CO3							x	
CO4								
CO5							x	
CO6							x	
CO7							x	
CO8							x	
CO9							x	

Mapping of course objectives to CIS student outcomes:

	Computer Information Systems Student Outcomes (CISSO)							
	x	2	3	4	5	6	7	8
CO1				x				
CO2							x	x
CO3							x	x
CO4								
CO5							x	
CO6							x	
CO7							x	
CO8							x	
CO9							x	

7. Brief list of topics to be covered
    1. Dangers of unethical hacking.
    2. Review of networks.
    3. Review of network security.
    4. Methods of reconnaissance.
    5. Methods for vulnerability scanning.
    6. Exploitation.
    7. Defensive programming basics.

1. Course number and name
   CSC 4102: Ethical Issues in Computer Science
2. Credits and contact hours
   1 credits / 12.5 hours.
3. Instructor's or course coordinator's name:
   Natasha Kurtonina, Frits Lander
4. Text book, title, author, and year:
   - Computer Ethics, Fourth Edition, by Deborah Johnson. ISBN-13: 978-0131112414 (required)
   - Ethical and Secure Computing: A Concise Module (Undergraduate Topics in Computer Science) 2nd Edition, by Joseph Migga Kizza , ISBN-13: 978-3030039363, ISBN-10: 3030039366

   a. other supplemental materials
      Notes, handouts and lectures are posted on Blackboard for student review.

5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course covers various ethical issues that arise as a result of increasing use of computers in contemporary society. Topics include methodology, tools and frameworks for analysis of ethical issues in Computer Science, social and ethical context of computing, professional and ethical responsibilities, intellectual property rights, risks and liabilities of safety-critical systems, privacy and civil liberties, social implications of the internet and cyber-security.
   b. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program:☒required; ☐elective; ☐selected elective.
6. Specific goals for the course:
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The purpose of this course is to give students a foundation in the structured programming with C and Unix Operating System that underlie the principles of process generation and process management, dynamic memory management, dynamic Data Structures, including Trees and Linked Lists, that will subsequently be used in Algorithms and Data Structures. (CSC3700). Upon completion of the course, a student should be able to do the following:
      - CO1: Identify ethical issues in Computer Science
      - CO2: Understand how to recognize and evaluate ethical choices in a modern computerized world.
      - CO3: Understand professional and ethical responsibilities defined in the ACM Professional Code of Ethics
      - CO4: Understand intellectual property rights, privacy and civil liberties, cyber-security, social and ethical implications of new technologies.
      - CO:5 Improve presentation skills
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes listed in Criterion 3

	CS Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1			Y	Y				
CO2			Y	Y				
CO3			Y	Y				
CO4			Y	Y				
CO5			Y	Y				

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	CIS Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1			Y	Y				
CO2			Y	Y				
CO3			Y	Y				
CO4			Y	Y				
CO5			Y	Y				

7. Major Topics Covered in the Course:
    a. Ethical frameworks for analysis of ethical issues in Computer Science.
    b. ACM Professional Code of Ethics: professional and ethical responsibilities in a computerized society
    c. Computer Security. Risks and liabilities of safety-critical systems.
    d. Intellectual Property, Privacy and Cybercrimes.
    e. The Digital Divide.
    f. Digital Identity and Digital Communities.
    g. Social Implications of Internet.
    h. Computer Crimes
    i. Ethical challenges arising from social networks, mobile telecommunications, virtual reality.
    j. Student’s term papers discussion

1. Course number and name
   CSC 4400 Software Engineering
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Nadimpalli Mahadev
4. Text book, title, author, and year
   Foundations of Software Engineering by A. Ahmed and B. Prasad, CRC Press 2016.
   a. other supplemental materials
      Class Notes and other handouts posted to Blackboard.
5. Specific course information
   a. brief description of the content of the course (catalog description)
      This course examines main features of software life cycle. It covers the main issues in design creation, principles of programs verification, system testing and evaluation criteria. This course provides students with an opportunity to obtain practical experience in software design using CASE tools. Students work in project teams and apply principles of software design, verification, testing and coding toward the solutions of assigned problems.
   b. prerequisites or co-requisites
      CSC 3011
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1: Professional and ethical responsibilities of a software engineer.
      CO2: Software Engineering processes and CASE tools.
      CO3: Requirements Analysis and documentation.
      CO4: Architecture design and documentation.
      CO5: Database design and documentation.
      CO6: User interface design and documentation.
      CO7: Object oriented analysis, design and documentation.
      CO8: Planning, documenting and implementing verification, validation and testing.
      CO9: Working in groups to complete a software project.

   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes:

	CS Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1				x			x	
CO2								
CO3						x	x	
CO4	x					x	x	
CO5						x	x	
CO6	x					x	x	
CO7	x					x	x	
CO8						x	x	
CO9			x		x		x	

Mapping of course objectives to CIS student outcomes:

	Computer Information Systems Student outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1				x			x	
CO2								
CO3		x					x	
CO4		x					x	
CO5		x				x	x	
CO6		x					x	
CO7		x					x	
CO8		x					x	
CO9		x	x		x		x	

7. Brief list of topics to be covered
   a) Software engineering overview and professional and ethical responsibilities (4 hours)
   b) Overview of systems engineering process (.5 hours)
   c) Overview of software processes and tools (3 hours)
   d) Project management overview (3 hours)
   e) Requirements engineering and the role of use cases (2 hours)
   f) Requirements documentation presentations (2 hours)
   g) Architecture models (1 hours)
   h) Data processing models and documentation (5 hours)
   i) User interface design and documentation (4 hour)
   j) Object oriented design and documentation (5 hours)
   k) Planning verification, validation and testing (3 hours)
   l) Software evolution and maintenance processes (3 hours)

1. Course number and name
   CSC 4550 Database Programming
2. Credits and contact hours
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name
   Nadimpalli Mahadev
4. Text book, title, author, and year
   No textbook.
   a. other supplemental materials
      Database Programming and Web Programming Notes.
5. Specific course information
   a. brief description of the content of the course (catalog description)
      Many businesses, small and large, utilize user-interfaces to access databases to provide security as well as multiple user access to the databases. This course will introduce the student to software components designed for database programming in a higher-level language. Each student works on a project of applied nature that involves documenting the requirements of a database, creating the database, and designing the user-interface to access the database. Working knowledge of the host language is required.
   b. prerequisites or co-requisites
      CSC 2400 and permission from the instructor
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Elective
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      Upon successful completion of this course, a student will have demonstrated knowledge of:
      CO1: Relational databases; entities, fields and records; relationships and keys.
      CO2: The process of designing a database.
      CO3: Components of a structured query language.
      CO4: Database security and data control.
      CO5: Three-tier architecture and MVC architecture.
      CO6: Analysis, design and implementation of an application using database programming

   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CS student outcomes:

	CS Student Outcomes (CSSO)							
	1	2	3	4	5	6	7	8
CO1								
CO2							x	
CO3							x	
CO4							x	
CO5							x	
CO6	x					x	x	

Mapping of course objectives to CIS student outcomes:

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1						x		
CO2						x	x	
CO3						x	x	
CO4						x	x	
CO5							x	
CO6		x				x	x	

7. Brief list of topics to be covered
    a. Relational database management systems.
    b. High level language (SQL)
    c. Transaction management and control commands
    d. Data security and data control commands
    e. Data entry and retrieve commands
    f. Database design.
    g. MVC and 3-tier architectures.
    h. Using JDBC and ODBC API in database programming.
    i. Design, analysis and implementation of applications using database programming.

1. Course number and name:
   CSC 4700 System Design and Implementation
2. Credits and contact hours:
   3 Credits, 37.5 Hours
3. Instructor's or course coordinator's name:
   Audrey Pereira
4. Text book, title, author, and year
   Systems Analysis and Design (9th ed.). by Kendall, K. E., & Kendall, J. E., 2014. Upper Saddle River, NJ: Pearson Education.
   a. other supplemental materials

5. Specific course information
   a. brief description of the content of the course (catalog description)
      This capstone course will emphasize the design and implementation phases of the system development life cycle. A problem-solving approach will be used for students to learn strategy and techniques for dealing with complexities in information systems development.
   b. prerequisites or co-requisites
      CSC 3710 Systems Analysis Methods
   c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
      Required
6. Specific goals for the course
   a. specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.
      The specific outcomes of instruction are address by the following course objectives:
      CO1: Use tools and describe steps required to design and implement good business systems
      CO2: Analyze good versus bad output and input designs
      CO3: Describe general guidelines for designing websites and mobile apps, including databases
      CO4: Evaluate different implementation options and describe approaches for the development of implementation plans, including test, training, roll-out, and security, security and privacy, and disaster recovery plans
      CO5: Discuss emerging trends and issues in systems design and implementation
      CO6: Work cooperatively in a group to integrate the concepts learned
      CO7: Construct and present effective oral and written forms of professional communications.
   b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

Mapping of course objectives to CIS student outcomes listed in Criterion 3

	Computer Information Systems Student Outcomes (CISSO)							
	1	2	3	4	5	6	7	8
CO1		X						
CO2		X						
CO3		X				X		
CO4		X						
CO5							X	
CO6					X			
CO7			X					

7. Brief list of topics to be covered
    a. Steps required to design and implement good business systems and available tools
        i. Human-computer interaction
        ii. Useability
        iii. User Interface types
        iv. User feedback
        v. Special design for ecommerce
        vi. Effective and efficient data capture
        vii. Input validation
    b. Designing effective output and input, and good versus bad output and input designs
        i. Output design objectives
        ii. Relating output content to output method
        iii. Realizing how bias affects users
        iv. Designing output for displays
        v. Good form design
        vi. Good display and web forms design
    c. Guidelines for designing websites and mobile apps, including databases
        i. Website design
        ii. Designing databases, including data concepts, normalization
        iii. Guidelines for master file/database relation design
        iv. Designing smartphone and tablet interfaces
        v. Designing queries
    d. Different implementation options and approaches for the development of implementation plans, including test, training, roll-out, and security, security and privacy, and disaster recovery plans
        i. Quality assurance and implementation
        ii. Testing approaches
        iii. User Training
        iv. Systems conversion and roll-out
        v. Security approaches and privacy concerns
        vi. Disaster recovery and planning
        vii. Evaluation techniques
    e. Emerging trends and issues in systems design and implementation
        i. Current events discussed as they emerge

APPENDIX B – FACULTY VITAE

1. Name
   Kevin B. Austin
2. Education – degree, discipline, institution, year
   - Ph.D., Biomedical Engineering, Worcester Polytechnic Institute (WPI), 1987
   - M.S., Engineering Science, Rensselaer Polytechnic Institute, 1985
   - B.S., Electrical Engineering, Polytechnic Institute of NY, 1980
3. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

Fitchburg State	Professor		2011-now	Full time
Fitchburg State	Professor	Graduate Chair	2012-15	Full time
Fitchburg State	Assoc. Professor		2006-11	Full time
Fitchburg State	Asst. Professor		2000-06	Full time
U. Colorado HSC	Regular Fellow		1991-98	Full time
McGill University	Research Associate		1987-91	Full time
WPI	Instructor		1985-87	Part time

4. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

ComputerBoards	Software Engineer	Software development	1999-2000	Full time
Eclectic Engineering Studio	President	Instrumentation engineer, consultant, media production	1998-now	Part time

5. Certifications or professional registrations

6. Current membership in professional organizations
   ACM (SIG Computer Science Education), IEEE Life Member (Engineering in Medicine & Biology, Computer, and Engineering Education societies)
7. Honors and awards
   - Principal Investigator, “Analysis of Echolocation Signals Obtained from Freely-behaving Bats in Natural Environments”, Fitchburg State Special Projects Grant (2014)
   - Principal Investigator, "Acquiring a Bat's Perspective on Biosonar Echoes", National Science Foundation, $576,463 (2008-2012)
   - Co-Principal Investigator, “Exploring the Ecology of Flash Communication in Photinus Fireflies Through Collaborative Undergraduate Research with Computer-Simulated Signaling Behavior”, National Science Foundation, $663,228 (2003-2006)
   - Co-Principal Investigator, “Social Impact of Information Technology”, Commonwealth Information Technology Initiative (2001)
   - NSERC International Fellowship in Science and Engineering. Department of Psychology, McGill University, Montreal, Quebec, Canada. (1991-93)
   - NSF-NATO Postdoctoral Fellowship in Science and Engineering. Department of Psychiatry, Douglas Hospital Research Center, McGill University, Montreal, Quebec, Canada. (1989-90)

- Research Institute of the Royal Victoria Hospital Postdoctoral Fellow. Department of Physiology, McGill University, Montreal, Quebec, Canada. (1988-89)
- Eta Kappa Nu. International Electrical Engineering honor society. Department of Electrical Engineering, Worcester Polytechnic Institute, Worcester, MA.

8. Service activities (within and outside of the institution)
   - Institutional Animal Care and Use Committee (IACUC, 2015-present)
   - CS and CIS Curriculum Committees
   - CS and CIS Search Committees (2014, 2016, 2017, 2018)
   - Departmental Peer Evaluation Committee
   - Baseball Coach (Jesse Burkett Little League, Worcester, MA)
   - Basketball Coach (Worcester JCC)
   - Robotics Coach (Midland Street School, Worcester, MA)
9. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation
   - Mahadev, N, K.B. Austin, "Sound localization by robot using inter-aural time differences." Journal of Computing Sciences in Colleges, 30(5):50-56 (April, 2015)
10. Briefly list the most recent professional development activities
    - Microchip Webinars:
      - Advanced Arduino Debugging (April 10, 2019)
      - Developing with AVR in MPLAB-X (March 26, 2019)
    - Xilinx Webinars:
      - ARM processors on Xilinx FPGAs. (January 24, 2019)
      - No hardware experience? No problem! Xilinx MicroBlaze processors are for everyone. (November 5, 2018)
    - Computer vision robotics / IoT project
      Developing a system for mobile robot tasking / game-playing based on a single video camera system broadcasting JSON objects to robotic platforms through an MQTT server.
    - Media production for education
      Experimenting with video presentation techniques for classroom demonstrations.
    - Bat echolocation signal processing
      My field research has yielded a large library of bat echolocation calls. Extracting significant echoes from is a challenging continuing research activity.

1. Name
   Xuzhou Chen
2. Education – degree, discipline, institution, year
   - MCS, Computer Science, North Carolina State University, Dec. 1998
   - Ph.D., Applied Mathematics, North Carolina State University, Dec. 1995
   - M.S., Numerical Analysis, Shanghai Teachers University, July 1987
   - B.S., Mathematics, Shanghai Teachers University, July 1984
3. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

Fitchburg State University	Full Professor (9/2013 –)	Department Chair	2013 – present	full time
Fitchburg State University	Associate Professor	Evening Program Chair (2008 – 2010)	2008 – 2013	full time
Fitchburg State University	Assistant Professor	Evening Program Chair (2003 – 2008)	2002 – 2008	full time
East China University of Science & Technology	Assistant Professor		1987 – 1991	full time

4. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

Nortel Networks	Member of Scientific Staff	R & D	1997 – 2002	full time
Fujitsu Network Communications	Software Engineer	R & D	5/1996 – 12/1996	full time

5. Certifications or professional registrations

6. Current membership in professional organizations
   ACM, CCSC, ILAS
7. Honors and awards
   - FSU Ruth Butler Grant to attend the 18th International Linear Algebra Society conference in Providence, RI, June 3-7, 2013.
   - FSU Special Research and Travel grants for research in Robotics Programming, 2012.
   - Fitchburg State University Center for Teaching and Learning CTL Innovation Grant: Enhancing Undergraduate Research and Learning of Advanced Algorithms using Robotics Programming, December 2010. (Joint with Dr. Mahadev)
8. Service activities (within and outside of the institution)
   - Member of Academic Advisory Board for CIS program at Mount Wachusett Community College.
   - Member of Academic Advisory Board for Montachusett Regional Vocational Technical School

9. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation
    - Lu He, Yan Luo, Rui Liu, Hengyong Yu, Xuzhou Chen, Seung Woo Son, Bisection and Twisted SVD on GPU, IEEE High Performance Extreme Computing Conference, Waltham, MA, 2015.
    - Jun Ji and Xuzhou Chen. A new method for computing Moore-Penrose inverse through Gauss-Jordan elimination. Applied Mathematics and Computation, Vol. 245, pp. 271-278, 2014.
    - Xuzhou Chen, Xinghua Shi, and Yimin Wei. The stationary iterations revisited. Numerical Algebra, Control and Optimization, Vol. 3, No. 2, 2013.
    - Yan Luo, Xuzhou Chen, and Jie Wang. A Virtual Network Embedding Algorithm for Providing Stronger Connectivity in the Residual Networks. Journal of Networks, Vol. 8, No. 4, April, 2013.
10. Briefly list the most recent professional development activities
    - Massachusetts STEM Summit 2018, November 14, 2018, DCU Center, Worcester, MA
    - National Initiative for Cybersecurity Education (NICE) 2018 Conference and Expo, November 6-7, 2018, Hyatt Regency Miami, Miami, Florida.
    - ABET Institute for the Development of Excellence in Assessment Leadership (IDEAL), Baltimore, July 31- August 3, 2017
    - 2016 Reconnect Workshop, Mathematical and Computational Tools of Cybersecurity, West Point, NY, June 12-18, 2016.
    - SIAM Conference on Applied Linear Algebra, Atlanta, GA, October 26-30, 2015. Presented Topic: Divide-and-Conquer Algorithm for Computing the Moore-Penrose Inverses
    - 2015 NSF National Workshop on Teaching an Undergraduate Parallel Programming Course with Pattern Programming in Washington, DC, July 12 –13, 2015.
    - The 19th International Linear Algebra Society (ILAS) conference, Seoul, Korea, August 3-7, 2014. Presented Topic: A new method for computing Moore-Penrose inverse through Gauss-Jordan elimination.
    - The 18th International Linear Algebra Society (ILAS) conference, Providence, RI, June 3-7, 2013.
      Presented Topic: The Stationary Iterations Revisited
    - Fitchburg State University Ruth Butler Award in 2013.
    - Fitchburg State University Special Projects Grants travel funds in 2012 and 2013.

1. Name
   Natasha Kurtonina
2. Education – degree, discipline, institution, year
   - Postdoctoral Study in Institute for Research in Language and Computation, 1997-1997 University of Pennsylvania
   - Ph.D., Computer Science University of Amsterdam and Utrecht University 1995
   - Ph.D., Logic, Moscow State University, 1990
   - B.S. and M.S, Mathematics and Philosophy Moscow State University, July 198
3. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

   Fitchburg State University 1999 – 2006 assistant professor
   2006 - associate professor
4. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
5. Certifications or professional registrations
6. Current membership in professional organizations
   ACM, CCSC, ILAS
7. Honors and awards
   Fitchburg State University Center for Italian Culture. Travel Grant to go to Italy to investigate some advantages of Computer Science Education (2006 together with Dr. Stephen Taylor)
8. Service activities (within and outside of the institution)
   - Member of Curriculum Committee
   - Member of the Graduate Committee
   - Member of many peer evaluations committees
9. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation
   - Natasha Kurtonina. Complexity Classes and Finite Model Theory for Substructural Logics. Journal of Logic, Language and Information (forthcoming)
   - Natasha Kurtonina. Relevant Logic Semantics Revised: Paraconsistency and Relevance. Logic Coloquium, University of Pennsylvania April 8, 2018 (Presentation)
   - Natasha Kurtonina. Categorial Grammar and Lambda Calculus. Logic and Language Seminar. European Summer School on Logic, Language and Computation. August 15, 2017 (Presentation)
   - Natasha Kurtonina. Computational Perspective on Substructural Logics. Logic in the Modern World Conference. Higher School of Economics, Moscow 2014 (Presentation)

10. Briefly list the most recent professional development activities
    - Developed all materials for undergraduate course “Game Programming”
    - Updated course material for “Programming Languages”
    - Developed all materials for a graduate class “Machine Learning”
    - Conferences and seminars
        - ✓ Harvard Logic Seminar (normally meets every other week)
        - ✓ UConn Logic Seminar (visit twice every semester)
        - ✓ American Association of Symbolic Logic. 2018 Winter Meeting Savannah Convention Center Savannah, Georgia January 3–5, 2018
        - ✓ American Association of Symbolic Logic. 2017 Winter Meeting Atlanta, GA, USA January 6–7, 2017

1. Name
   Frits Lander
2. Education – degree, discipline, institution, year
   - Artium, Science and Mathematics, Rødovre State School Denmark, June 1992
   - Academy Engineer, Mechanical Engineering, Engineering Academy of Denmark, July 1968
   - Postgraduate Courses, Computer Science, Technical University of Denmark, 1971 - 1978
3. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

Fitchburg State University	Associate Professor		2013 – present	full time
Fitchburg State University	Associate Professor	Department Chair	2004 – 2013	full time
Fitchburg State University	Associate Professor		1993 – 2004	full time
Fitchburg State University	Assistant Professor		1986 – 1993	full time
Fitchburg State University	Instructor		1982 - 1986	full time

4. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

North East Systems Associates	Consultant	Software engineering of Scope54, a semi-automated package designed to fully support and enhance the performance of the HP 54120 digitizing oscilloscope.	1989 – 1995	part time
Digital Equipment Corporation	Consultant	Consulting the design and construction of an EMI measurement site.	1988	part time
AVCO Systems Division	Consultant	Simulation of conical dipole antennas.	1983	Part time

5. Certifications or professional registrations
   Oil Burner Technician Certificate, BU-030664
   Commonwealth of Massachusetts, Department of Fire Services
6. Current membership in professional organizations

7. Honors and awards
    - Faculty Member of the Year 2003.
    - Merit Bonus, December 2002.
8. Service activities (within and outside of the institution)
    - Member of Academic Advisory Board for Montachusett Regional Vocational Technical School
9. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation
10. Briefly list the most recent professional development activities
    - ABET Symposium, April 19-21, 2012, St. Louis, MO
    - ABET Symposium, April 14-16, 2011, Indianapolis, IN

1. Name
   Nadimpalli V. Mahadev
2. Education – degree, discipline, institution, year
   - Ph.D., Combinatorics & Optimization, University of Waterloo, Canada, 1984
   - M.Math., Combinatorics & Optimization, University of Waterloo, 1980
   - B. Stat (Hons)., Mathematics and Statistics, Indian Statistical Institute, Kolkata, 1978
3. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

Fitchburg State University	Professor		2002 – present	full time
Fitchburg State University	chair		2000 - 2004	full time
Fitchburg State University	Associate Professor		1999 – 2002	full time
Northeastern University	Associate Professor		1992 – 1999	full time
Northeastern University	Assistant Professor		1986 – 1992	full time
University of Winnipeg	Assistant Professor		1984 – 1986	full time

4. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time
5. Certifications or professional registrations
6. Current membership in professional organizations
   ACM SIGCSE
7. Honors and awards
8. Service activities (within and outside of the institution)
   - Served on college-wide committees such as Information Technology Advisory Board, Promotions Committee, Academic Policies Committee etc.
   - Devised the assessment plan for the two undergraduate programs in the department.
   - Served as a volunteer chief judge in Spelling Bee, Geography Bee etc., organized by North South Foundation.
9. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation
   - Building a Secure Hacking Lab in a Small University, Conference Paper, ITiCSE 2017.
   - Developing a Holistic Understanding of Systems and Algorithms through Research Papers, with A. Erkan et al, Conference Paper, ITiCSE 2017.
   - Sound localization by robot using inter-aural time differences, with K. Austin, article, Conference Paper CCSE 2015.

10. Briefly list the most recent professional development activities
    - Suggested Cyber Security Concentration in CIS program.
    - Developed a course in Ethical Hacking.

1. Name
   Ricky J. Sethi
2. Education – degree, discipline, institution, year
   - Ph.D., Computer Science, University of California, Riverside, 2009
   - M.S., Physics/Business (Information Systems), University of Southern California, 2001
   - B.A., Molecular and Cellular Biology, Neurobiology (Physics minor), University of California, Berkeley, 1996
3. Academic experience – institution, rank, title (chair, coordinator, etc. if appropriate), when (ex. 1990-1995), full time or part time

Fitchburg State University	Associate Professor		2018 – present	full time
Fitchburg State University	Assistant Professor		2014 – 2018	full time
Southern New Hampshire University	Team Lead and Adjunct Professor		2013 – present	part time
University of California, Los Angeles University of Southern California - Information Sciences Institute	NSF Computing Innovation Fellow		2010 – 2013	full time

4. Non-academic experience – company or entity, title, brief description of position, when (ex. 1993-1999), full time or part time

The Madsci Network	Director of Research	R & D	2013 – Present	part time
US National Academy of Sciences (NAS)'s The Science and Entertainment Exchange	Consulting Scientist	R & D	2018 – Present	part time

5. Certifications or professional registrations
   • Member, American Institute of Physics
   • Fellow, North American Academy of Arts and Sciences
   • Member, YSP/Madsci Financial Board
6. Current membership in professional organizations
   ACM, IEEE, AIP
7. Honors and awards
   - Amazon (AMZN) 2017 – 2019, $35,000
     "Structured Discussions and Scientific Workflows for Data Analysis"
   - National Endowment for the Humanities (NEH) 2016 – 2017,. $40,000

"Scientific Workflows, Image Analysis, and Visual Stylometry in the Digital Analysis of Art"

- National Science Foundation (NSF) 2010 – 2012, $285,371
  "Machine Learning Framework for Social Computing and Collective Intelligence"

8. Service activities (within and outside of the institution)
   - NSF Panelist, NSF Scalable Data CyberInfrastructure and NSF Cyberlearning, 2012-2018
   - Co-Chair, Making WAIVS: Workflows for the Analysis of Images in Visual Stylometry
   - Co-Chair, Scientific Workflows for Machine Learning Applications
9. Briefly list the most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation
   - Richard De Veaux, Mahesh Agarwal, Maia Averett, Benjamin Baumer, Andrew Bray, Thomas Bressoud, Lance Bryant, Lei Cheng, Amanda Francis, Robert Gould, Albert Y. Kim, Matt Kretchmar, Qin Lu, Ann Moskol, Deborah Nolan, Roberto Pelayo, Sean Raleigh, Ricky J. Sethi, Mutiara Sondjaja, Neelesh Tiruviluamala, Paul Uhlig, Talitha Washington, Curtis Wesley, David White, and Ping Ye, "Curriculum Guidelines for Undergraduate Programs in Data Science". Annual Review of Statistics and Its Application (Annu Rev Stat Appl) (2017).
   - Ricky J. Sethi and Yolanda Gil, "Scientific Workflows in Data Analysis: Bridging Expertise Across Multiple Domains". Future Generation Computer Systems (FGCS) (2017).
   - Ricky J. Sethi, Crowdsourcing the Verification of Fake News and Alternative Facts". ACM Conference on Hypertext and Social Media (ACM HT) (2017).
10. Briefly list the most recent professional development activities
    - Interview, Future of Artificial Intelligence, Newsy Television, 2018. https://www.youtube.com/watch?v=VU-fVl0pqNM
    - Keynote Speaker, Fact-Checking via Structured Discussions in Virtual Communities, 3rd International Workshop on Social Media World Sensors (Sideways), Prague, Czech Republic, 2017.
    - Editorial Board Member, International Journal of Computer Vision & Signal Processing, 2011 - Present

APPENDIX C – EQUIPMENT

Laboratory facilities are available for student use in conjunction with various courses. Some courses require special equipment and/or software. Here, equipment and software are listed by room. Every classroom on campus is technology-mediated with an instructor podium and projector. Our department has membership in the Microsoft Developer Network Academic Alliance which allows us to deliver many development tools to them for free via a DreamSpark e-commerce site (http://msdnaa.fitchburgstate.edu).

E201 and 202 labs (used for CS 1, CS 2, CS Basics, Application Programming, Systems Programming, Programming Languages, Object Oriented Programming, Algorithms and Data Structures)
24 thin client computers running Windows and the following software
Microsoft Visual Studio
Microsoft Office
Java, Netbeans
Eclipse
XAMPP for MySQL and TomCat
Python
Perl
SWI Prolog
Scheme
Microsoft Project
Microsoft Visio
Dia diagramming software
Greenfoot
Textedit
Anaconda
Putty
Raptor
DOSbox
Logisim

E207 lab (used for Microprocessors, Computer Organization, Data Communications, Game Programming, Parallel Programming, Mobile Applications)
8 Lab Stations running Windows 7 (Nvidia graphics cards)
Unity, Visual Studio, MPLabX, CCS PICC, Tera Term, Arduino IDE, Android Studio, Xilinx Vivado, Logisim, Multisim.
Tektronix Digital Sampling Oscilloscopes
Saleae Logic Analyzers
Lab Benches
Prototyping systems
PICkit Microcontroller Programmers
Digilent Analog Discovery II USB Instrument

Digilent CMOD Artix 7 FPGA modules
LVSim-Dcomm Digital Communications Simulation Software

E203 lab (used for Introduction to Electronics and Digital Electronics)
12 Lab Stations Microsoft Windows 7
Multisim circuit simulation software
Tektronics Digital Sampling Oscilloscopes
Circuit prototyping systems w/ power supplies, signal generators, digital inputs/outputs
Digital Multimeters
MPlabX IDE
Xilinx Vivado
Digilent Analog Discovery II USB Instrument
Digilent BASYS-3 Artix 7 FPGA modules

E102 (used for Ethical Hacking)
21 Computers used for Ethical Hacking lab

E205 Project Lab
3 – Matlab programming/data analysis workstations
3 – USB data acquisition subsystems
1 GHz digital sampling oscilloscope
Network analyzer
Logic Analyzer
Sherline CNC Milling machine
Sherline CNC Lathe
Drill press
Band saw
Printed circuit board (PCB) fabrication materials
Ultiboard PCB layout software

Other (virtual)
Linux server used in various programming classes
Oracle database server

APPENDIX D – INSTITUTIONAL SUMMARY

1. The Institution

a. Fitchburg State University 160 Pearl Street
Fitchburg, MA 01420

b. President of the university is Dr. Richard Lapidus and the Vice President for Academic Affairs is Dr. Alberto Cardelle.

c. This report is submitted by the Dean of Business and Technology, Dr. Keith Williamson.

d. Fitchburg State University is accredited by the New England Commission of Higher Education (formerly known as the New England Association of Schools and Colleges, Inc.). The most recent accreditation evaluation visit was in 2012.

2. Type of Control

Fitchburg State University is one of six comprehensive State Universities in the Massachusetts higher education system; and like its sister institutions, the University's governance procedures and organizational structure are largely mandated by either state law or collective bargaining agreements. The official governing body is the Board of Higher Education (BHE) which is staffed by the state's Department of Higher Education (DHE), led by the Commissioner of Higher Education. The BHE consists of the Secretary of Education, ex officio, or his designee, thirteen voting members appointed by the governor to reflect regional geographic representation, and three members chosen to represent public institutions of higher education.

According to the Department of Higher Education's website, the Board has four fundamental responsibilities:

1. Define the mission of and coordinate the Commonwealth's system of higher education. The Department, in conjunction with the universities' Boards of Trustees, holds the system accountable for achieving its goals and establishing a comprehensive system to measure quality by defining educational achievement and success with the use of standards and measurements.
2. Approve the awarding of degrees and define and authorize new functions new programs or consolidate, discontinue, or transfer existing functions, educational activities, and programs.
3. Analyze present and future goals, needs, and requirements of public higher education and establish goals to achieve a well-coordinated quality system of public higher education in the Commonwealth.
4. Develop a rational and equitable statewide tuition plan for state universities and the community colleges.

In addition, the Board, through the BHE, establishes policy for state colleges and universities and outlines performance measures to use as comparisons with peer institutions in other states. As the employer of record for all those represented by the four collective bargaining agreements, the BHE is responsible for collective bargaining negotiations, contract, and grievance administration (see http://www.mass.edu/about/aboutdhe.asp). Contracts and collective bargaining agreements are available on the Office of Human Resources website. (See https://www.fitchburgstate.edu/offices-services-directory/human-resources- payroll/collectivebargaining/)

Massachusetts law delineates the roles of the Board of Trustees and the chief operating officers of the state universities. Individual boards of trustees for Massachusetts higher education institutions were established by legislation in 1980.
In accordance with the laws of the Commonwealth and regulations enacted by the Massachusetts Board of Higher Education, the composition, duties and powers of the State University Board of Trustees are articulated within the university's bylaws. The board is charged with the fiduciary management of the institution, including determination of fees, establishment of personnel management policy, staff services, and the general business of the institution. Among its responsibilities, the board elects the president with the approval of the Board of Higher Education, adopts an annual plan of financial operation, awards degrees in approved fields, and develops the mission statement for the university consistent with the mission of the Commonwealth's system of public higher education.

The board of trustees consists of 11 voting members. Nine are appointed by the governor for five-year terms, renewable once; one alumni trustee is elected by the Alumni Association for a five-year term; and a student trustee is elected by the student body for one year. No member may serve for more than two consecutive terms.

The officers consist of a chairman, vice chairman and clerk. They are elected by the trustees following nominations by a committee. The president of the university serves in a non- voting capacity to the board (adapted from https://www.fitchburgstate.edu/offices-services-directory/board-of-trustees/) .

There have been many significant structural changes at our institution during the period since our last accreditation visit, including the hiring of a new President, Dr. Richard Lapidus and a Provost, Dr. Alberto J.F. Cardelle, the creation and implementation of a new administrative structure including academic deans, and the creation of a new office dedicated specifically to institutional research, planning and data-informed decision-making. Also, Fitchburg State University recently completed its five-year strategic plan, which is guiding our efforts going forward. (adapted from NEASC 5th year interim report: https://www.fitchburgstate.edu/offices-services-directory/academic-affairs/neasc-5th-year- interim-report/)

The roles and responsibilities of each administrator are defined and kept on file, along with the procedures for their selection, in the office of the Assistant Vice President for Human Resources and Payroll Services. Currently, the President is advised by an Executive Cabinet (EC) composed of the Vice President for Academic Affairs, Vice President of Student Affairs, Vice President for Finance and Administration, Vice President for Institutional Advancement, Chief Information Officer and Assistant Vice President, Assistant Vice President, Human Resources/Payroll Services, Associate Vice President for Academic Affairs, Executive Director of Marketing and Integrated Communications. In addition to his meetings with the EC, the President meets frequently with the Vice President for Academic Affairs and the Vice President for Finance and Administration. The EC also meets regularly; and each of these administrators holds regular staff meetings. The Vice President for Academic Affairs meets weekly with the academic affairs administrators. (adapted from https://www.fitchburgstate.edu/offices-services-directory/administration/)

The structure of academic governance at Fitchburg State University, as in the other state universities, is governed by the contractual agreement between the BHE and the Massachusetts State College Association (MSCA). The duties and responsibilities of department chairpersons are defined by contract; they are elected by department faculty in accordance with procedures established by Article VI of the contract and may serve up to three consecutive three-year terms. In addition to meeting with department faculty, the chairs meet at least monthly with the Academic Dean and, at Fitchburg State, all department chairs meet monthly on their own. These separate meetings are intended to facilitate communication with the Academic Vice President and focus on developing agenda items for upcoming Vice President/Chairs meetings.

A separate contractual agreement between the BHE and the MSCA lists the responsibilities of the University and the faculty teaching in Graduate and Continuing Education (GCE).
GCE programs are administered by graduate program chairs and evening undergraduate program managers; these are positions created by the University and filled by the Graduate and Continuing Education Dean, in consultation with department chairs, who also meets with her appointees several times a year.

At the departmental level, governance begins with curriculum committees which meet to consider new course offerings, program revisions and new program development. Approved proposals, along with those made by faculty, students, or administrators and those relating to changes in academic or student life policies are then submitted to the All University Committee (AUC). The AUC is composed of eight faculty members elected by their peers, three administrators appointed by the President, and three students selected by the Student Government Association (SGA). The AUC has three standing committees -- Curriculum, Academic Policies, and Student Affairs--as well as ad hoc committees to deal with special issues. The Curriculum and Academic Policies committees are composed of 16 faculty

members appointed by the MSCA, three administrators appointed by the President, and three students selected by the SGA, while the Student Affairs committee is comprised of nine students, five administrators and five faculty members. The standing committees recommend approval or disapproval to the AUC which then makes a recommendation of approval or disapproval to the President who has the final authority on each change.

There is a separate, contractually defined governance structure for graduate policies and curriculum. Departments with a graduate program also have a graduate curriculum committee which may recommend changes to graduate curriculum or policies. If both the graduate program chair and the department chair approve the changes, they are reviewed by the Graduate Council. The Graduate Council then makes a recommendation of approval or disapproval to the President who has the final authority on each change. The Graduate Council is composed of five faculty members, appointed by the MSCA, three administrators appointed by the President, and one graduate student elected by the other Graduate Council members.
At the beginning of each academic school year the President hosts two opening addresses, one for classified personal and administrators and the other for faculty, librarians, and administrators. These addresses typically review major past events and forecast those expected in the new academic year.

Every other year, the graduate program coordinators, undergraduate program managers, and graduate and continuing education faculty meet at the beginning of the academic year. These meetings include a series of professional development workshops, departmental meetings, and updates from the Graduate Dean, Vice President for Academic Affairs, and President.
Additionally, the Dean and Associate Dean of Graduate and Continuing Education meet with graduate chairs and undergraduate program managers throughout the year as needed. Once each semester, there is a combined meeting for all department chairs, graduate chairs, and undergraduate program managers with the entire academic affairs team for the purpose of discussing items of common interest.

3. Educational Unit

Brady Chen is the chair of the department. As of July1, 2019, Dr. Nadimpalli Mahadev will be assuming the chair of the department. The chair reports to Dr. Keith Williamson, the Dean of Business and Technology. The deans report to the Vice President of Academic Affairs, Dr. Alberto Cardelle, who reports to President Richard Lapidus.

4. Academic Support Units

The chair of the Mathematics Department is Dr. Jenn Berg.
The chair of the Geo/Physical Science Department is Dr. Elizabeth Gordon.

5. Non-academic Support Units

The head of the library is Dean Jacalyn Kremer. The
Chief Information Officer is Steven Swartz.
The Tutoring Center is run by Director Chris Coffin.
Career Counseling and Advising Center is headed by Director Erin Kelleher.
Counseling is headed by Robert Hynes.
Disability Services is run by Director Katrina Durham.

6. Credit Unit

One semester or quarter credit generally represents one class hour or three laboratory hours per week. One academic year normally represents a minimum of 28 weeks of classes, exclusive of final examinations. In general, 3 hours credit equals 135 Carnegie Units; with each Carnegie unit representing 9 hours per week for 15 weeks.

7. Tables

Table D-1. Program Enrollment and Degree Data

Name of the Program: Computer Science

	Academic Year		Enrollment Year					Total Undergrad	Total Grad	Degrees Awarded			
			1st	2nd	3rd	4th	5th			Associates	Bachelors	Masters	Doctorates
Current Year		FT								0			0
		PT											
2018		FT	33	26	22	66		147	42	0	8	27	0
		PT	4	2	8	20		34	15				
2017		FT	28	19	34	58		139	38	0	12	27	0
		PT	1	2	3	18		24	23				
2016		FT	33	24	29	54		140	43	0	4	25	0
		PT	0	0	6	15		21	13				
2015		FT	33	16	27	49		125	37	0	13	35	0
		PT	0	0	3	16		19	24				

Give official fall term enrollment figures (head count) for the current and preceding four academic years and undergraduate and graduate degrees conferred during each of those years. The "current" year means the academic year preceding the on-site visit.

FT--full time
PT--part time

Table D-2. Personnel

Name of the Program: Computer Science

Year[1]: 2018-2019

	HEAD COUNT		FTE[2]
	FT	PT	
Administrative[2]	0.375	0	0.375
Faculty (tenure-track)[3]	5 .875	0	5.875
Other Faculty (excluding student Assistants)	0	0.833	0.833
Student Teaching Assistants[4]	0	0	0
Technicians/Specialists	0	0	0
Office/Clerical Employees	1	0	1
Others[5]	0	0	0

Report data for the program being evaluated.

1. Data on this table should be for the fall term immediately preceding the visit. Updated tables for the fall term when the ABET team is visiting are to be prepared and presented to the team when they arrive.

2. Persons holding joint administrative/faculty positions or other combined assignments should be allocated to each category according to the fraction of the appointment assigned to that category.

3. For faculty members, 1 FTE equals what your institution defines as a full-time load

4. For student teaching assistants, 1 FTE equals 20 hours per week of work (or service). For undergraduate and graduate students, 1 FTE equals 15 semester credit-hours (or 24 quarter credit-hours) per term of institutional course work, meaning all courses — science, humanities and social sciences, etc.

5. Specify any other category considered appropriate, or leave blank.

APPENDIX E – FOUR-YEAR STUDY PLAN

Computer Science

FRESHMAN YEAR

Fall Semester		*16 Credits*
ENGL 1100	Writing I	(3)
CSC 1500	Computer Science I	(3)
MATH 1300	Precalculus (if needed)	(3)
PHYS 2300	General Physics I	(4)
	LA&S Elective	(3)

Spring Semester		*16 Credits*
ENGL 1200	Writing II	(3)
CSC 1550	Computer Science II	(3)
CSC 1900	Discrete Math	(3)
PHYS 2400	General Physics II	(4)
	LA&S Elective	(3)

SOPHOMORE YEAR

Fall Semester		*17 Credits*
CSC 1600	Introduction to Electronics	(4)
CSC 2560	Systems Programming	(3)
MATH 2300	Calculus I	(4)
	LA&S Elective	(3)
	LA&S Elective	(3)

Spring Semester		*17 Credits*
CSC 1650	Digital Electronics	(4)
CSC 3700	Algorithms and Data Structures	(3)
MATH 2400	Calculus II	(4)
SPCH 1000	Introduction to Speech Communication	(3)
	LA&S Elective	(3)

JUNIOR YEAR

Fall Semester		*16 Credits*
CSC 2600	Computer Organization	(4)
CSC 3200	Programming Languages	(3)
MATH 1800	Business Statistics	(3)
CSC 3xxx/4xxx	CSC Elective	(3)
	Free Elective	(3)

Spring Semester		*16 Credits*
CSC 3600	Microprocessors	(4)
CSC 3100	Operating Systems	(3)
MATH 2600	Linear Algebra	(3)
CSC 3xxx/4xxx	CSC Elective	(3)
	Free Electvie	(3)

SENIOR YEAR

Fall Semester		*12 Credits*
CSC 3011	Data Modeling and Database Design	(3)
CSC 3xxx/4xxx	CSC Elective	(3)
	LA&S Elective	(3)
	Free Elective	(3)

Spring Semester		*10 Credits*
CSC 4400	Software Engineering	(3)
MATH 3xxx	Math Elective	(3)
CSC 3xxx/4xxxx	CSC Elective	(3)
CSC 4102	Ethical Issues in Computer Science	(1)

**Suggested courses for Math minors; all others may substitute free electives.*

LA&S Elective List

1 AOM attribute (Art or Music)
1 CTW attribute (Citizenship & The World)
3 credits HAF attribute (Health/Fitness)
1 HIST subject (History)
1 HMN attribute (Human Behavior)
1 LIT attribute (Literature)

Advanced LA&S Options Area

Option B requirements fulfilled by completion of MATH 1800, 2300, 2400, and 2600.

Global Diversity Area

Two courses taken must meet the Global Diversity requirement: GDAN course + (GDC or GDCN course) ***OR*** *GDCN course + (GDA or GDAN course). These courses are allowed to satisfy this requirement and another requirement at the same time.*

Suggested Computer Science/ Mathematics Electives:

Fall Semester
CSC 3050 Web Programming with Java
CSC 3400 Data Communication and Networking
CSC 3500 Object Oriented Programming
CSC 4940 Internship: Computer Science

Spring Semester
CSC 3300 Assembly Language CSC
3450 Local Area Networks CSC
4550 Database Programming
CSC 4940 Internship: Computer Science
MATH 3000 Geometry

Completion of 120 credits required forgraduation.

Game Programming

FRESHMAN YEAR

Fall Semester — *16 Credits*

ENGL 1100	Writing I	(3)
CSC 1500	Computer Science I	(3)
MATH 1300	Precalculus (if needed)	(3)
PHYS 2300	General Physics I	(4)
History	LA&S Elective (CTW)	(3)

Spring Semester — *16 Credits*

ENGL 1200	Writing II	(3)
CSC 1550	Computer Science II	(3)
CSC 1900	Discrete Math	(3)
PHYS 2400	General Physics II	(4)
EXSS 1000	Health and Fitness	(3)

SOPHOMORE YEAR

Fall Semester — *17 Credits*

CSC 1600	Introduction to Electronics	(4)
CSC 2560	Systems Programming	(3)
MATH 2300	Calculus I (Option A)	(4)
	LA&S Elective (LIT)	(3)
	LA&S Elective (CTW)	(3)

Spring Semester — *17 Credits*

CSC 1650	Digital Electronics	(4)
CSC 3700	Algorithms and Data Structures	(3)
MATH 2400	Calculus II (Option B)	(4)
SPCH 1000	Introduction to Speech Communication	(3)
Behavioral Science	LA&S Elective (CTW)	(3)

JUNIOR YEAR

Fall Semester — *16 Credits*

CSC 2600	Computer Organization	(4)
CSC 3200	Programming Languages	(3)
MATH 1800	Business Statistics (Option B)	(3)
GAME 2000	Elements of Game Designs	(3)
CSC 3650	Game Programming	(3)

Spring Semester — *16 Credits*

CSC 3600	Microprocessors	(4)
CSC 3100	Operating Systems	(3)
MATH 2600	Linear Algebra (Option B)	(3)
CSC 3560	Mobile Application Developments	(3)
	Free Elective	(3)

SENIOR YEAR

Fall Semester — *12 Credits*

CSC 4100	Computer Graphics	(3)
CSC 3100	Data Modeling and Database Design	(3)
GAME 2200	Introduction to Game Art	(3)
	LA&S Elective (AOM)	(3)

Spring Semester — *12 Credits*

CSC 4400	Software Engineering	(3)
MATH 3xxx	Math Elective	(3)
CSC 3xxx	CSC Elective	(3)
GAME 3000	Game Design Workshop	(3)

LA&S Elective List

1 AOM attribute (Art or Music)
3 CTW attribute (Citizenship & The World)
1 LIT attribute (Literature)

Advanced LA&S Options Area

Option B requirements fulfilled by completion of MATH 1800, 2300, 2400, and 2600.

Global Diversity Area

Two courses taken must meet the Global Diversity requirement: GDAN course + (GDC or GDCN course) **OR** *GDCN course + (GDA or GDAN course). These courses are allowed to satisfy this requirement and another requirement at the same time.*

Suggested Computer Science/ Mathematics Electives:

Fall Semester
CSC 3050 Web Programming with Java
CSC 3400 Data Communication and Networking
CSC 3500 Object Oriented Programming
CSC 4940 Internship: Computer Science
GAME 3060 3D Game Development

Spring Semester
CSC 3300 Assembly Language CSC
3450 Local Area Networks CSC
4550 Database Programming
CSC 4940 Internship: Computer Science

Completion of 120 credits required forgraduation.

Rev. 9-2018

APPENDIX F – LIBRARY SERVICES

The New England Commission on Higher Education's Standard 7.22 calls for "access to library and information resources, services, facilities, and qualified staff sufficient to support its teaching and learning environments and its research and public service mission as appropriate." The purpose of this report is to outline the current Amelia V. Gallucci-Cirio Library's resources, services and facilities that support the undergraduate program in Computer Science at Fitchburg State University. After review of the data compiled for this report, the library resources, services, and facilities are deemed sufficient to meet the needs of undergraduate researchers in Computer Science and to support faculty in their teaching.

ABOUT Computer Science at Fitchburg State University

Total Computer Science students enrolled in Fall 2018	100
Total Computer Information Systems students enrolled Fall 2018	64
Total Students	164

An analysis of the library support needed for the Computer Science undergraduate major are classified into three categories: resources, services and facilities.

RESOURCES for Computer Science

Researchers in Computer Science use both academic journals and monographs (books). Currency of sources is critical with an emphasis on the prioritization of sources less than 5 years old. Faculty in Computer Science are also interested in incorporating streaming films/documentaries into their teaching.

1. Journals and Databases

The Amelia V. Gallucci-Cirio Library offers access to over 100,000 online journals in over 165 databases. Specifically for the Computer Science major, the Library purchases the databases Applied Science & Technology Source, ACM Digital Library, and ProQuest Computing. These databases are excellent sources of literature for computer science studies. Applied Science & Technology Source provides full-text content from more than 840 journals and magazines. Usage statistics show some declining usage of this database over the past four years, although the database is still well used. ACM Digital Library provides a full-text collection of all ACM publications, including journals, conference proceedings, technical magazines, newsletters, and books. Since 2014 ACM Digital Library has seen a significant increase in usage. ProQuest Computing provides full-text and peer reviewed content from over 500 journals and has seen steady usage in the most recent years. See Library Table 1: Full-text Journal Databases by Disciplines related to Computer Science**.**
Looking at Computer Science through an interdisciplinary lens, the library offers numerous business and communication related databases and these databases adequately meet the needs of an undergraduate researcher. See Library Table 1.

The library collection development policy has been and continues to be to provide the core journals and databases for each discipline. Reviews of databases and journals are consulted, peer comparisons are conducted, and faculty input on the effectiveness of the resource is critical when considering new databases. Funds for new databases and/or journals are then requested, and if granted, they are purchased. In fall 2018, the library conducted a journal review project. It looked at the approximately 400 print and online journals to which the library subscribes (outside of the journals available through the databases). The library determined the annual cost per usage by dividing the annual cost for the journal title by the number of times the journal was used in a year. Criteria was established and applied that allowed the library to cancel journals that were not being effectively used. This allowed the library to increase journal offerings in needed areas as determined by interlibrary loan data as well as to purchase a large, multi-disciplinary eBook collection. More information about the new eBook collection is below.

2. Books

A review of our print collection in the Library of Congress call number ranges specifically associated with Computer Science shows approximately 7,100 books in our collection. This is just below the optimal depth of collection. See Library Table 2: Monograph Collection Description and Analysis.

In addition, almost all the books were in the print collection as the Library offered few eBooks. It was our recommendation that an eBook package that includes computer science books be acquired to meet the needs of the undergraduate researcher and the faculty**.** This

would not only increase the number of volumes available, it would also

increase the number of books published in the past 5 years. Therefore, effective March 2019, the EBSCO Academic Complete eBook package was subscribed to that included approximately 12,190 computer science related books to meet the needs of the undergraduate researcher and the faculty**.** Over 1600 of these books were published in the last 5 years and 6,045 were published in the past 10 years. This increases the number of books associated with Computer Science in total to over 19,250 books while providing off-campus access. This total number brings the collection well above the advanced level for undergraduates (over 12,000 books) to the Advanced support level.

3. Films and other Media

In 2018, the Library purchased a subscription to the academic streaming film database Kanopy. Over 250 videos are available with subject headings aligned with Computer Science. See Library Table 3: Films and Other Media Collection for a breakdown by category.

SERVICES for Computer Science

Library Instruction

For all academic departments in the 2018 academic year, faculty librarians taught over 184 research sessions and were embedded into 63 courses. Through these efforts, we reached over 4,340 students during the last academic year. With only 6 faculty librarians on staff, the number of classes with research sessions and/or an embedded librarian is impressive and requests continue to increase.

There were no research sessions taught for computer science courses. The library would like to have conversations with computer science faculty to explore how the library can support the research needs of computer science students through our instruction program, particularly in research-intensive courses. See Library Table 4: Research Instruction for more information.

Library Research Guides

The Library offers 32 subject research guides plus over 100 course specific guides, covering all disciplines at Fitchburg State. For Computer Science, we have created 1 subject research guide and 1 course specific research guide. The usage statistics for the Computer Science research guide show the guide was accessed 21 times in a year, about one fifth the usage the average subject guide receives. Starting in fall 2019, the Library's Computer Science Research Guide will be made available at point-of-need within the Blackboard course management system in all Computer Science courses in order to facilitate access.

Research Help

The Library offers one-on-one reference services in a variety of modes, including dropping in at the reference desk, making a personal appointment, email, and chat instant messaging service. The overwhelming majority of such services are offered in-person at the research help desk, although this number has declined significantly over time. During the academic year, research help is available to students for 60 hours per week. The aggregate trends in research help appear in Library Table 5: Research Help. Statistics on the use of research help by Computer Science students only are not available.

Reserves

The Library's Reserve system is well used by the Fitchburg State community. For example, this semester 98 professors put a total of 567 items on reserves. Checkouts of reserve materials by all students were more than 2,000 during the last academic year. For the past three semesters, there have been no Computer Science items on Reserves. In our fall, 2019 meeting with Computer Science faculty we will be discussing the opportunity for them to utilize Reserves. In addition, the Library is currently exploring ways to increase access to materials by Computer Science students, including the insertion of digital library resources into courses and the adoption of Open Educational Resources in order to give students more access to no or low cost textbooks and other course materials.

Interlibrary Services Request

Data shows Computer Science students and professors rarely use Interlibrary Loan Services. As a department, they ranked among the lowest on their use compared to other departments. See Library Table 6: Interlibrary Services for details.

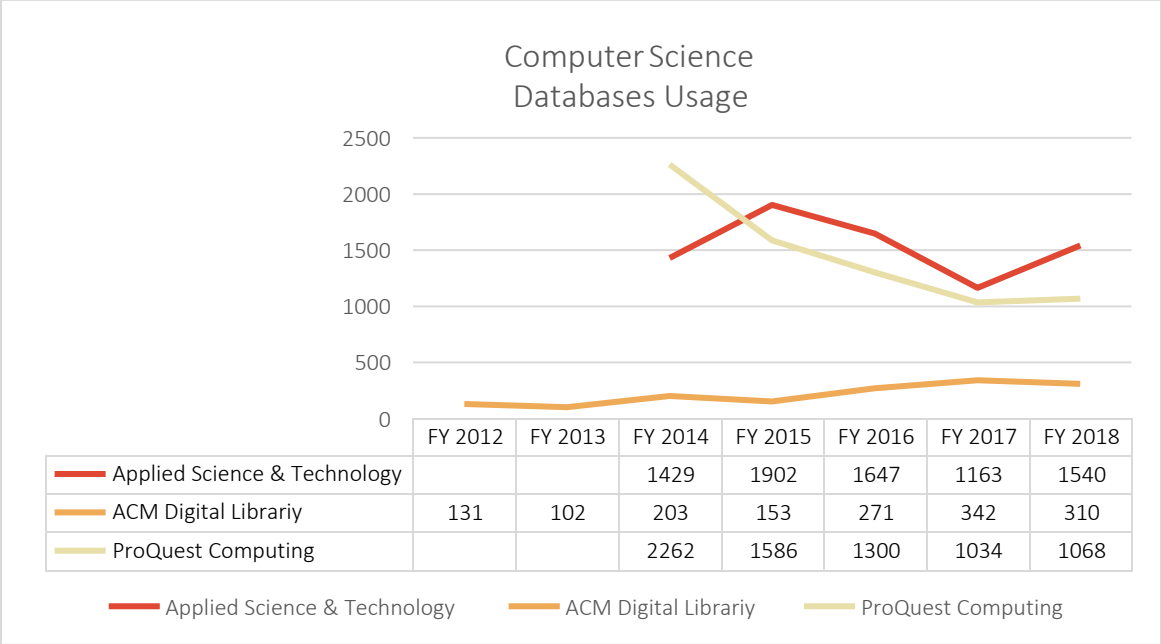
FACILITIES for Computer Science

With the Library's recent renovation, students have access to welcoming spaces designed to support individual and group work. Building information is in Library Table 7: Facilities. The full complement of databases treating Computer Science can be found on the library website (https://library.fitchburgstate.edu/research/databases/computer-science/). Whereas there are 5 directly applicable full-text databases, another 11 full-text databases supplement this core collection. In addition, individual titles that stretch across the sciences and that are embedded within databases or that we subscribe to separately (and which are accessible through Serials Solutions) number in the thousands.

Core Full-text Journal Databases
1. Applied Science and Technology Source 2. ProQuest Computing 3. ACM Digital Library 4. ProQuest Science Database 5. Science & Technology Collection
Supplemental Full-text Journal Databases
1. ABI/Inform 2. Academic OneFile 3. Academic Search Ultimate 4. Business Source Premier 5. Credo Reference 6. Directory of Open Access Journals 7. Expanded Academic ASAP 8. GPO Access/FedSys 9. Library Information Science & Technology Abstracts 10. MathSciNet 11. SpringerLink

Library Table 1:
Full-text Journal Databases by Disciplines related to Computer Science

Database usage data disaggregated by discipline does not exist and it is not possible to determine how many articles were accessed by computer science faculty and students only. In total though for the Fitchburg State community, over 155,000 articles were accessed through the Library's 165 databases in fiscal year 2018.



Library Table 2:
Monograph Collection Description and Analysis

Fitchburg State University is, by Carnegie classification, a Master's granting institution. The Computer Science program offers a B.S. degree, and therefore the University must currently uphold at least the standard of 3a, "Basic Study" for its collections, with the goal of offering the standard of 3b, "Intermediate Study" (see below).

General Guidelines for Monograph Collection Depth

1 Minimal – A level that consists mostly of basic works.
2 Basic Information.
2a A level that introduces and defines the subject and that indicates the varieties of information available elsewhere.
2b Basic Instructional Support – A level that introduces course work and research for undergraduate courses, including a wide range of basic monographs and reference tools pertaining to the subject and targeted to undergraduate students.
3 Study or Instructional Support.
3a Basic Study – A level that supports undergraduate courses.
3b Intermediate Study – A level that supports upper division undergraduate courses.
3c Advanced instructional Support – A level that supports course work and research for graduate and undergraduate courses, including a wide range of basic monographs and reference tools pertaining to the subject.
4 Research – A level that supports independent research and preparation of doctoral dissertations
5 Comprehensive Inclusion – Comprised of all significant works for a defined topic.

*Specific Definitions for Monograph Holdings**

1b (or less) Minimal level = less than 2,500.
2a Basic introductory level = 2,500 - 5,000 titles.
2b Basic advanced level (Community College) = 5,000 - 8,000 titles.
3a Instructional support (lower level undergraduate) = 8,000-12,000 titles representing a range of monographs.
3b Intermediate support level (advanced undergraduate) = more than 12,000 titles representing a wider range than 3a.
3c Advanced support level (Master's degree level) = more than 12,000 titles representing a wider range than 3c.

*Quantitative WLN Criteria for Determining CL

Computer Science Book Collection

LC Subject Area Computer Science	LC	2012	2013	2014	2015	2016	2017	2018
Computer Games. Video Games. Fantasy Games.	GV1469	-	2	2	2	2	2	2
Business Apps.	HD30.2	54	52	55	55	59	59	58
E-business	HD30.38	-	2	2	2	2	2	2
Ethical Hacking	HD8039	-	110	113	114	115	118	118
Computer Industry	HD9696	94	94	95	95	96	97	97
Office Automation	HF5548	3	3	2	3	3	3	3
Computer Programming, Hacking	HM851	-	48	59	61	63	70	76
Computer Crime, Internet Security	HV6773	-	18	19	19	20	22	22
Educational Uses	LB1028	43	38	37	38	38	38	42
Mathematics	QA	1,751	2,137	2,369	2,524	2,674	2,843	2,969
Machine Theory	QA150- 272	484	259	280	283	301	324	341
Analysis	QA300- 433	388	222	230	232	239	249	259
Computer Science/EDP	QA75- 76.95	1,519	746	891	1,001	1,073	1,153	1,207
Physics	QC	1,196	961	1,000	1,014	1,047	1,062	1,088
Game Design/Computer Graphics, Graphics Programming	T385	33	34	52	53	55	57	58
Operations Research, Information Systems, Systems Analysis	T58	4	2	2	2	2	2	2
Misc.	TA156 & ZA	0	10	11	14	15	23	23
Data Processing	TA1630-1660	25	19	22	24	26	27	27
Telecommunications	TK1501-6720	707	459	485	504	519	543	558
Electronics and Computer Engineering	TK7807-7895	594	173	179	180	184	186	190
Total		6895	5389	5905	6220	6533	6880	7142

In the period under review, the monograph collection in computer science experienced steady growth, 3.58% from 2012-2018. Like other disciplines, the computer science collection has undergone extensive weeding. In December 2016 a major weeding project was completed to eliminate obsolete computer science monographs.

The total number of print books in the call number ranges associated with Computer Science is 7,142. This is just below the number expected for a collection to support lower level undergraduate (8,000-12,000 books) and advanced level undergraduate (over 12,000 books). Effective March 2019, the EBSCO Academic Complete eBook package was subscribed to that included approximately 12,190 computer science related eBooks to meet the needs of the undergraduate researcher and the faculty**.** Over 1600 of these eBooks were published in the last 5 years and 6,045 were published in the past 10 years. This increases the number of books associated with Computer Science in total to over 19,250 books while providing off-campus access. This total number brings the collection well above the advanced level for undergraduates (over 12,000 books) to the Intermediate support level.

Library Table 3: Film and Other Media Collection

# of Streaming Films by Subject in Kanopy Database	
Computer Science & Technology	254
Cybercrime	23
Cybersecurity	23
Video Gaming	9
Robotics	28
Computers	8
Business Technology	7
Tech Industry	10
Big Data	7
Internet	19
IT Business Skills	27

Library Table 4: Library Instruction

	AY2012	**AY2013**	**AY2014**	**AY2015**	**AY2016**	**AY2017**	**AY2018**
Total Instruction Sessions Conducted:	166	211	197	161	222	263	247
Computer Science Sessions Conducted:	0	0	0	0	0	0	0
Percentage	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Total Embedded:	3	13	16	18	42	99	63
No. of CS Embedded:	0	0	0	0	0	0	0
Total One-shots:	163	198	181	143	180	164	184
No. of CS One-shots:	0	0	0	0	0	0	0

** The library offers discipline-specific and general information literacy instruction sessions.*

Library Table 5: Research Help

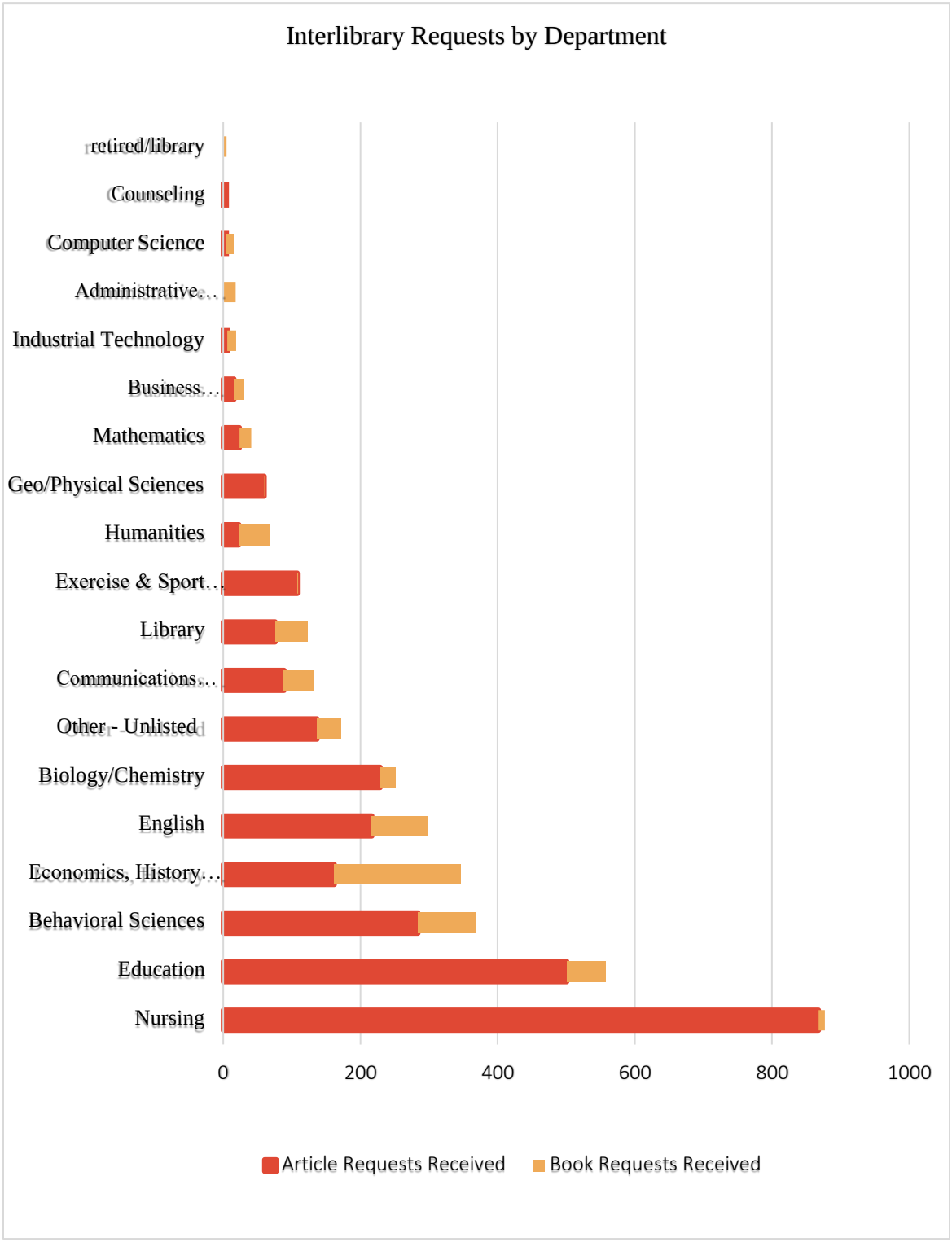
Library Research Guides

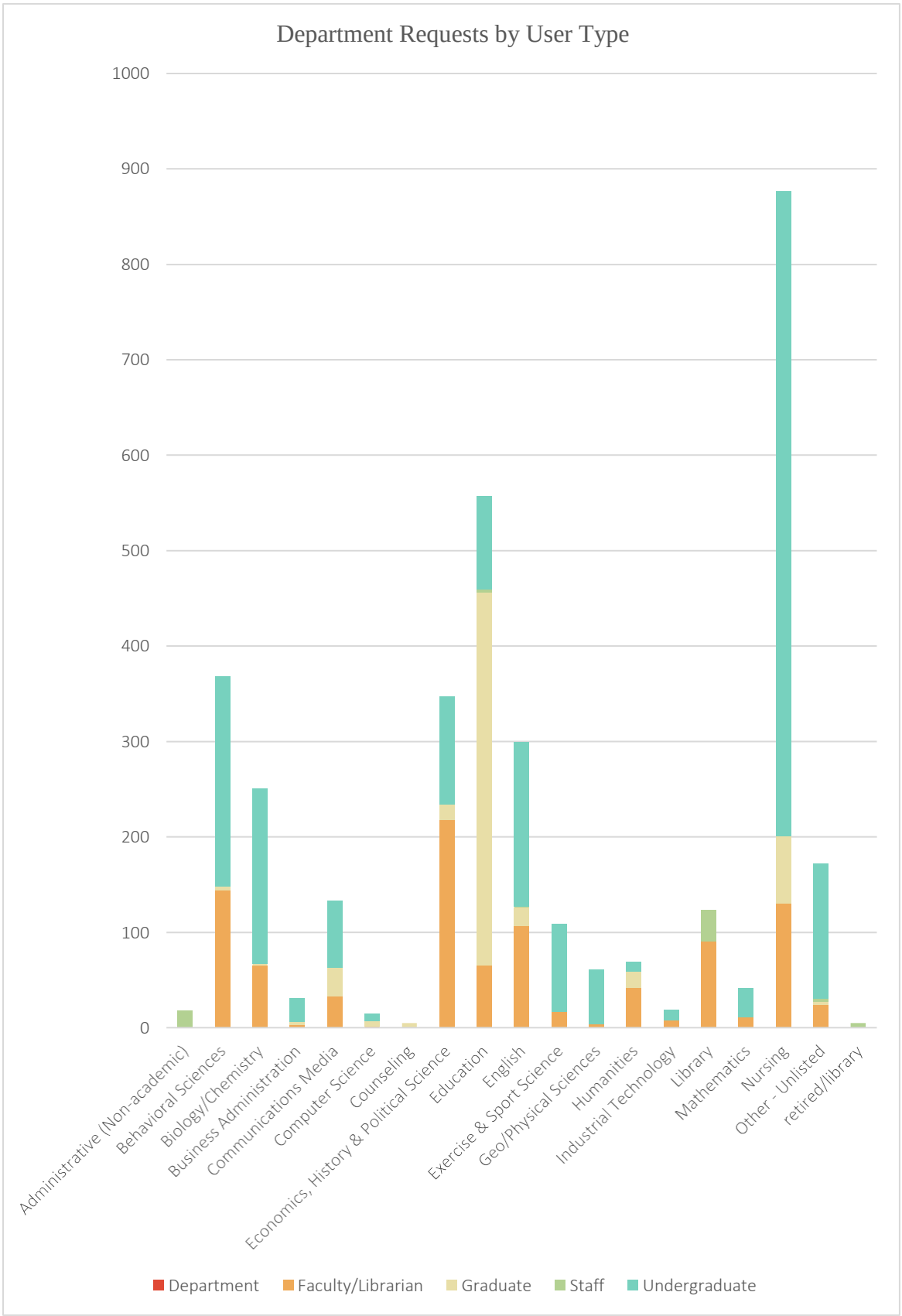
For Computer Science, we have created one subject research guide and one course specific research guide. The usage statistics in the Computer Science research guide show the guide was accessed 21 times in a year, about one fifth the usage the average subject guide receives.

Reference Statistics for University

	FY2012	FY2013	FY2014	FY2015	FY2016	FY2017	FY2018
Total Records	2957	4377	3544	2642	2497	1875	2854
Mode of Access	FY201 2	FY2013	FY2014	FY2015	FY2016	FY2017	FY20 8
In Person	2037	3383	2490	1959	1872	1386	2297
Chat	728	779	678	548	510	308	268
Phone/Email	178	133	272	133	112	162	287
Skype	0	0	12	0	0	0	1
Office/Appointment	14	82	47	2	3	19	0
Blackboard	0	0	27	0	0	0	10
Questions by Patron	FY2012	FY2013	FY2014	FY2015	FY2016	FY2017	FY2018
Student	1671	3426	3016	2438	2320	1674	2632
Faculty	75	104	102	59	66	57	65
Extended Campus/DL	32	15	256	27	21	45	112
Public/Alumni/Other	140	165	145	111	79	89	131
Staff	15	29	16	7	11	10	8
Unknown	1024	638	13	0	0	0	0
Duration	FY2012	FY2013	FY2014	FY2015	FY2016	FY2017	FY2018
0-2 minutes	995	1510	1449	1104	1006	782	1483
2-5 minutes	465	1215	1008	735	683	532	689
5-15 minutes	895	1079	628	509	424	327	331
15 minutes or longer	448	466	466	294	384	234	351
Blank	155	107	1	0	0	0	0

Library Table 6: Interlibrary Services




Department Requests by User Type
1000
900
800
700
600
500
400
300
200
100
0
Administrative (Non-academic)
Behavioral Sciences
Biology/Chemistry
Business Administration
Communications Media
Computer Science
Counseling
Economics, History & Political Science
Education
English
Exercise & Sport Science
Geo/Physical Sciences
Humanities
Industrial Technology
Library
Mathematics
Nursing
Other - Unlisted
retired/library
Department
Faculty/Librarian
Graduate
Staff
Undergraduate

Library Table 7:
Facilities

Space	Specifications
Total Number of Seats in Library	596
Information Commons	Research Help Desk Circulation Desk 61 public computer stations 3 multi-function printers.
Study Rooms	9 large (up to 8 people) containing conference table, white board, media viewing equipment, and Apple TV. 8 small (2 people) containing conference table, computer, and whiteboard.
Media Production Room	Seating up to 7 people containing a computer, Apple TV, ceiling mounted projector, DVD player, and document projector.
Quiet Space	2 floors (3rd and 4th)
Archives	8608 items used in FY18 38 Special Collections totaling 322 boxes. 13 record groups totaling 480 boxes 2,500 rare books Art collection

Study Room Statistics	FY18
Unique Users	1758
Total Bookings	9203
Hours Booked	1630.5

SUBMISSION ATTESTING TO COMPLIANCE

Only the Dean or Dean's Delegate can electronically submit the Self-study Report.

ABET considers the on-line submission as equivalent to that of an electronic signature of compliance attesting to the fact that the program conducted an honest assessment of compliance and has provided a complete and accurate disclosure of timely information regarding compliance with ABET's *Criteria for Accrediting Engineering Programs* to include the General Criteria and any applicable Program Criteria, and the ABET *Accreditation Policy and Procedure Manual.*