COLORADO MESA UNIVERSITY

AY 2016 – 2017
Program Review

Computer Science

Computer Science Program Review
AY 2016-2017

Contents

1. Introduction and Program Overview

1.a. Program description by level, identifying concentrations and minors as applicable;

The Computer Science program is within the department of Computer Science, Math & Statistics. It supports a BS and AS degrees, as well as a CS minor and Computer Security certificate. Additionally, the program provides an Essential Learning course (CSCI 100 Computers in our Society) and courses in support of Mechanical Engineering (CSCI 130 Computer Science for Engineers). The introductory courses are also used by the Teacher Education and Mathematics programs.

Our staff has gone through significant adjustment during this program review. With major changes for more than ½ the staff as is summarized in the following table.

Name	Rank	Start	Notes
Lori Payne	Professor	< 2000	½ time, also CSMS department chair
Gary Rader	Professor	< 2000	
Warren MacEvoy	Professor	2001	
Ram Basnet	Assistant Professor	2013	
Karl Castleton	Assistant Professor	2014	Transitioned from ¾ instructor to tenure track.
John Dobbs	Instructor (0.8)	2016	
Arun Ektare	Professor	< 2000	Retired 2014, transitional 2013
Anne Spalding	Associate Professor	2002	Did not return from Sabbatical started 2012

Part-time faculty teach less than 10% of our FTE but represent important diversity in resources for the program and our students.

The main thread of the program is to provide a background for students interested in working in the field either directly (our primary goal) or after additional graduate school. The dynamics and scope of the field, which substantially reinvents itself in one academic program review cycle, is an ongoing challenge.

1.b. Brief history of the program;

In 1974, the campus transitioned from a 2-year to a 4-year school, with a mission to offer non-traditional four-year degrees. Thus was initiated the offering of a 4-year degree in Computer Science; at the time, we were the only institution in the state of Colorado that offered a 4-year degree (as opposed to a graduate degree) in Computer Science. The original degree was offered successfully for several years. In 1980, a baccalaureate degree with an emphasis in computer science replaced the earlier program. Finally, with the change from emphasis areas to majors and minors in 1992, Mesa State College offered a Bachelor's of Science degree in computer science. In the 24 years since its adoption, 250 B. S. in Computer Science degrees have been awarded.

1.c. Recommendations from the previous external review and progress made toward addressing them (copy found on CMU assessment website);

Recommendation # 1: Recruitment and retention of computer science students – ongoing efforts must be made to retain the students we have. It should be noted that in the last program review, the retention rate was 41.2%, while it was 46.1% this time. Efforts have evidently improved retention somewhat, but further efforts must be made.

> Response: We have improved retention by more than 10%. We have made efforts to better place students through advising both before and after they declare as majors. We have also introduced tutoring within our labs and aids during courses. We have also expanded our connection to our students through involvement in three student organizations: Association for Computing Machinery (professional computer science society, advised by Warren MacEvoy), Computer Security Club (advised by Ram Basnet and started in 2015) and UPE (computer science honor society, advised by Ram Basnet).

Recommendation # 2: Developing online courses – Since MSC offers some online degrees, all general education courses need to be available online as well. A commitment has been made to begin offering the CSCI 100 and CSCI 106 courses in the next year, but other coursework may also be suitable for online delivery.

> Response: CSCI 100 Computers in our Society has regularly been offered online, which is now an Essential Learning course, and other courses have been offered online, such as CSCI 106 Web Page Design I, CSCI 206 Web Page Design II, and CSCI 333 Unix.

Recommendation # 3: Support for faculty research and development – With the high cost of seminars and conferences which examine cutting edge information in Computer Science, faculty often feel that faculty development funds are inadequate. The average available to a faculty member is $600, which rarely covers even the conference fee. To circumvent this, we often agree as a group to pool these funds or roll them over in order that some of our faculty can participate in conferences and workshops.

> Response: Our participation in the DARPA Robotics challenge shows we are involved in cutting edge world-class research projects. The support we have had through the campus and department greatly facilitated that effort which is still bringing rewards to the department (examples are the NASA space grant work and AgBot robotic farming challenge, as well as notability in the region). We have also been able to expand our server room and support dozens of student

research projects thanks to the support of the campus in supporting our facility and special travel needs. The faculty has also taken advantage of external funding such as NSF Security in Education (SEED) grants for security workshops and ACM funding for bringing external speakers and conference travel. We hope the campus continues to support these efforts.

Recommendation # 4: Combining CSCI 241 Computer Architecture and CSCI 321 Assembly Language into a single course with a lab based component. The Assembly Language course is challenging for many of our majors, yet is not a large part of the field today. Still, the faculty feel the course is important for the goals of our program – a well-rounded individual familiar with both hardware and software. Moving this material as a portion of our computer architecture diminishes the depth of coverage (which is reasonable considering its place in the current field), but still introduces them to concepts they may need for embedded systems designs.

Response: Done. CSCI 321 used to be one of the most difficult courses for students to pass, even though the skills were typically only indirectly applicable. The pass rate is much better in the combined class without greatly diminishing the technical background of our students.

Recommendation # 5: Enable the CSCI 310 Advanced Programming course to be taught in 1-3 hour sections, and then offer one or two of these per semester. Many not in computer science ask for language classes to be taught; some not big enough to justify a 3 credit hour course. This will allow the short courses, and also allow courses to be taught in J-term, or in short summer sessions whereas a 3 hour course might not. Also, other departments have asked us to teach such courses (such as Flash for the Graphic Arts/Animation majors) for their students; this change will allow us to do that.

Response: Done.

Recommendation # 6: All programs (AS, BS, and Minor in CS) will be required or allowed to take CSCI 310 for credit. The committee felt that students need to take language classes instead of trying to incorporate them in courses already filled with material.

Response: Done.

Recommendation # 7: Changing the CS Minor to include alternates to the CSCI 241 Architecture course, so that those not interested in hardware can obtain a minor. Options added include the new course added for Engineering (CSCI 130) or CSCI 206 Web Page Design II. This change should also encourage the students interested in web programming to get a minor in CS.

Response: Done.

Recommendation # 8: Change the AS in Computer Science to include the CISB 205 Advanced Business Software, the CSCI 106 and CSCI 206 Web Page Design I and II, or the CSCI 130 Intro to Engineering CS courses. This would allow students in the AS program to take courses which could help them get jobs in the field, making it a more desirable program.

Response: Done.

Recommendation # 9: Add new courses to the CS choice list to reflect changes in the field. Courses added are popular new courses CSCI 370 Computer Security and CSCI 345 Video Game Programming.

Response: Done.

Recommendation # 10: Increase the number of courses available online – Besides being able to generate some FTE by offering such courses, general education courses need to be offered online to meet MSC's commitment to the online degree programs. Additionally, online programs are more accessible to professionals currently working in technical fields and wanting to stay abreast of current computing technologies.

Response: We now offer our Essential Learning courses online. The availability of workshops in collaboration with the Maverick Innovation Center has been discussed, but a full format of the workshops have not yet been determined.

Recommendation # 11: Provide more mentoring/tutoring to lower-level computer science students – Already in progress, hiring lab assistants/tutors with the course fee money has proven to be successful but must be taken further. The faculty have voted to create a special group of students, called the Mourey Group in honor of a past faculty member/mentor of many of us here now, to aid directly in the classroom and outside for those courses identified as causing students difficulty. A group of three has been selected for this year, and their use is already in place this semester.

Response: Done.

1.d. Mission statement and goals for the program, including the program's centrality to CMU's role and mission and strategic plan, and as applicable, how it adds value to the region;

We have no program mission statement, other than to be a strong regional program in Computer Science in alignment with the campus mission statement:

> Committed to a personal approach, Colorado Mesa University is a dynamic learning environment that offers abundant opportunities for students and the larger community to grow intellectually, professionally, and personally. By celebrating exceptional teaching, academic excellence, scholarly and creative activities, and by encouraging diversity, critical thinking, and social responsibility, CMU advances the common good of Colorado and beyond.

Goals and related sub-objectives are the following.

Goal 1: Provide a strong program in computer science.

a. Help students obtain an understanding of computer hardware, and the relationship between hardware and software, including security implications.

b. Help students develop proper problem analysis techniques and programming skills.

c. Provide students with strong logical and critical thinking skills.

d. Develop ethical skills with respect to security and privacy.

Goal 2: Prepare students for employment in such fields as software engineering and design, system programming, applications programming, and cybersecurity.

a. Provide a curriculum that evolves to meet the demands of the industry.

b. Provide students with an internship / real world experience in their field prior to graduation

c. Encourage independent learning skills.

d. Help students develop professional skills needed in the work environment.

Goal 3: Prepare students to enter graduate programs in computer science.

a. Provide a curriculum that meets standards of major graduate programs.

b. Provide opportunities for research projects and develop research skills.

Goal 4: Provide an opportunity for persons in the geographic region to increase their skills and employability.

a. Provide an Associate's degree curriculum that focuses on the demands of industry.

b. Increase the number of people proficient in computer science to address the nationwide need for technical literacy.

1.e. How the program's curriculum supports other majors/minors and general education requirements, as applicable

Outside Essential Learning, our most popular cross-departmental course is CSCI 130 Introduction to Computer Science which is taken by incoming engineering freshman.

1.f. Locational/comparative advantage;

The only other CS program in the western slope is a very small program at Western State Colorado University which is roughly both our CISB and CSCI programs with 1.5 faculty.

1.g. Any unique characteristics of the program; and

Our decision to maintain embedded language skills (C/C++) as a core reference language is unique in the state and gives our students access to high-performance/embedded/security career paths. This is reflected in local employment with engineering firms and large employers like Visa and Hill Air Force Base who are specifically interested in our students.

As a regional resource; we are the unique path to a well compensated (90% employment wth average salary 40-60k) and rewarding career that is consistently in the top 10 choices for jobs for the state and nation (see BLS statistics in Appendix C)

1.h. Other information/data (program's option).

None.

2. Curriculum

2.a. Describe the program's curriculum in terms of its breadth, depth, and level of the discipline.

Our program tends toward the applied; again with the expectation that most of our students would like to work directly after their degree. For example, we have not offered a theory of computation in many years course other than as an independent study. We offer limited courses in other specializations, such as computer graphics (which is only offered once every other year). Major additions since our last program review are computer security (a new program in 2016) and embedded systems. The success of our students in the workforce and as graduate students, the alignment our courses with other universities in our area, and to the ACM/IEEE guidelines indicate a strong and evolving program that strives to maintain relevance in the field.

We offer a fairly traditional approach to the degree. From a mathematical background we ask baccalaureates to complete MATH 152 Calculus 2 and MATH 369 Discrete Structures as well as STAT 200 Introduction to Statistics. Our associate program only requires MATH 113 College Algebra (although students are strongly encouraged to continue in order to transition into the BS degree).

Within the field, the sequence starts with CSCI 111 CS1 and CSCI 112 CS2 which introduce students to the basic structure of a computer, data structures, and a beginning of how to program in C++. This continues with CSCI 250 CS 3. These three together give our students a fairly good introduction to a core language (C++), suitable as a reference point for other languages (such as Java or C#) and topics (such as operating systems and security). This core sequence is finished with CSCI 241 Computer Architecture before students need to pick areas of interest.

From here there are two principle groups of classes: required upper division courses and an elective list of upper division courses. The required courses are CSCI 330 Programming Languages, CSCI 470 Operating System Design, CSCI 484 Compute Networks, and CSCI 490 Software Engineering. The last is our capstone course. These courses represent core knowledge we feel all CS students must have before graduating, and are standards across CS programs.

The directed elective list (15 hours) gives students some flexibility in pursing their interests in the field. These include various topic areas: CSCI 306 Web Page Design III, CSCI 322 Embedded Systems, CSCI 333 UNIX Operating Systems, CSCI 337 User Interface Design, CSCI 345 Video Game Design, CSCI 370 Computer Security, CSCI 375 Object Oriented Programming, CSCI 380 Operations Research, CSCI 445 Computer Graphics, CSCI 450 Compiler Structure, CSCI 460 Database Design, CSCI 480 Theory of Algorithms, CSCI 486 Artificial Intelligence, MATH 361 Numerical Analysis. This list is under review and we plan on adding CSCI 445 Mobile Applications and CSCI 405 Computer Security to the directed elective list.

Finally, we require 5 hours in a set of programming languages grouped as 310 Advanced programming. These hours are designed to be flexible, as the industry retools, but currently Java, Python, and C# are regular offerings, given as 1 to 3 credit hours depending on content. This is to address a breadth of languages fundamental to software engineering, which almost never involves only one language.

2.b. Program currency. What curricular changes have been made since the last program review?

CSCI 100 Computers in Our Society is now an Essential Learning course.

CSCI 104 Computer Hardware was added to improve students' hardware background.

CSCI 130 Introduction to Computer Science for Engineers was added in support of the Mechanical Engineering degree.

CSCI 331 Assembly Language was merged into CSCI 241 Computer Architecture to streamline the course sequence and de-emphasize very low-level programming.

CSCI 345 Video Game Design was added due to interest, market size and general applicability.

CSCI 405 Mobile Applications was added in support of new devices.

CSCI 370 Computer Security was added for the security certificate.

CSCI 420 Cyber Security was added for the security certificate.

CSCI 464 Network / Application Security added for the security certificate.

A Certificate track in Computer Security was added to the program in 2016. We also changed the organization of the associates and minor so that MATH 152 Calculus 2 is not required (among other changes) so that students that struggle with higher level mathematics can still complete an associates or certificate.

We have merged CSCI 321 Assembly Language with CSCI 241 Computer Architecture, we added CSCI 405 Mobile Apps, CSCI 104 Computer Hardware, CSCI 130 Introduction to Computer Science for Engineers.

2.c. Description of program delivery locations and formats and how it has shifted to meet the changing needs of its students.

We offer CSCI 100 Computers in our society and CSCI 106 Introduction to Web Page Design online. We also offer these classes (as well as CSCI 310 Advanced Programming) in the evening to increase student availability.

While cost prohibitive to offer the full degree online, we have worked to accommodate specific student needs. For example, creating a hybrid online version of a course for a student that could not attend all classes in person, working with a student that was deployed over a 2 semester gap, or a remote independent study for a student working in Honduras. Such arrangements are specific and not uncommon.

3. Analysis of Student Demand and Success

Our program in 2009 was in jeopardy; the collapse of the field c. 2001 was hard felt academically. Even though the job market had recovered, student perception had not and programs around the country shrank. Since then we have seen modest steady growth. Outlook from the Bureau of Labor Statistics both for the state and nationally consistently rank careers in computer science as in-demand, financially rewarding and among the best options for a 4-year degree from the point of view of educational investment. We believe these trends will continue.

Locally, our most important change was the hiring of Ram Basnet; his background in Computer Security allowed us to significantly expand our offerings in this very important career area, and we now have a healthy cadre of students working through the certificate program. Karl Castleton's involvement with CS education at the K-12 area is also bearing fruit, since these students are now entering our program with much better background compared to the past, thanks in large part to the efforts of Karl Castleton supporting FIRST Robotics (a combined team across all district high schools that did not exist in 2009) and FIRST LEGO (there are now over 30 from zero in 2009) leagues within School District 51.

An interesting local decision is regarding language. Many universities use Java as the principle language for their students, while we use C++, which is a more general and more difficult language. As a consequence, our students seem to have better access to embedded application development. For example, we have a significant number of students working at Visa which must manage many transactions quickly. This background is also interesting locally because of engineering companies which are local. For example Biorad labs does large and fast data analysis related to inverse spectroscopy, and so desires engineers skilled in C++ to develop fast algorithms. This language specialization gives our students employment opportunities many other campuses do not match and distinguishes our program from others in the state. Our ability to produce students with core systems, embedded and security skills gives us a unique value and story that we believe will continue to attract students.

3.a. Number of majors (by concentration(s)) and minors;

Computer Science Major Profile, Fall 2011-2015

		2011	2012	2013	2014	2015
Degree	Major	#	#	#	#	#
AS	Liberal Arts-Computer Science	38	36	31	40	31
PB	Computer Science Prov Bacc		4	11	12	18
BS	Computer Science	109	119	119	124	149

		2011		2012		2013	
Degree	Major	Avg Index	# Scores	Avg Index	# Scores	Avg Index	# Scores
AS	Liberal Arts-Computer Science	84	13	84	17	87	13
PB	Computer Science Prov Bacc			79	3	84	11
BS	Computer Science	105	80	103	89	106	94

		2014		2015	
Degree	Major	Avg Index	# Scores	Avg Index	# Scores
AS	Liberal Arts-Computer Science	81	23	81	18
PB	Computer Science Prov Bacc	85	12	86	18
BS	Computer Science	106	102	106	120

3.b. Registrations and student credit hours by student level;

		2011-12		2012-13		2013-14		2014-15		2015-16	
Subject	Student Level	1st	All	1st	All	1st	All	1st	All	1st	All
CSCI	Freshmen	145	460	149	493	108	366	131	440	111	378

Sophomore	235	750	225	722	236	764	223	723	272	859
Junior	155	492	142	440	167	531	150	478	197	603
Senior	333	1007	382	1128	345	1017	327	988	356	1025
PBL	1	4	1	3	2	6	1	3	0	0
Graduate	1	3	0	0	0	0	1	3	0	0
Graduate Non-Deg	0	0	0	0	0	0	0	0	3	11
Non Degree	0	0	4	12	3	10	2	6	2	7
	870	2716	903	2798	861	2694	835	2641	941	2883

3.c. Registrations and student credit hours (fall and spring terms) subtotaled by course level;

		2011-12		2012-13		2013-14		2014-15		2015-16	
Subject	Course Level	1st	All	1st	All	1st	All	1st	All	1st	All
CSCI	100	502	1623	478	1556	454	1501	479	1564	495	1575
	200	96	319	93	309	74	243	98	329	103	347
	300	148	402	216	585	210	581	178	508	221	595
	400	124	372	116	348	123	369	80	240	122	366

3.d. Number of graduates (by concentration);

Degrees Awarded, Academic Year

Degree	Major	2010-11	2011-12	2012-13	2013-14	2014-15
AS	Liberal Arts-Computer Science		3	2	3	3
BS	Computer Science	7	15	16	23	21
Total		7	18	18	26	24

3.e. Student successes/recognitions, especially in external student competitions;

Brandon Foss, Emilee Castleton, Michaela Ervin, Luke Schaefer, Robert VanMatre, Sophat Sem, Eric Wilcox, and Sam VanPelt participated in the international DARPA Robotics Grand Challenge. Josh Garland obtained his PhD in Applied Math and is a Omidyar Fellow at the Santa Fe Institute. Dan Crumly obtained his PhD in Computer Science and now works at the Snow and Ice data center in Boulder. Marshall Sweatt and Eric Wilcox finished their Master's programs in Computer Science. Eric's specialty started in Robotics but is now Computer Security, while Marshall works with Robotics in the Arab Emirates.

3.f. Other information/data (program's option).

None.

4. Program Resources

4.a. Faculty

4.a.1. Ratio of full-time equivalent students (FTES) to full-time equivalent faculty (FTEF);

See table 9.

4.a.2. Course credit hours and student credit hours by faculty type (i.e., tenured/tenure-track, instructor, administrators/staff/coaches, lecturers);

See table 8.

4.a.3. Faculty successes/quality/recognitions - details related to teaching, advising, scholarship, service, and other achievements;

Teaching: New courses in Mobile Applications, Robotics and Cyber Security, as well as a Cyber Security Certificate Track. Many courses have gone through major revisions in the past five years.

Service: CSMS Department Chair, Graduation Committee Chair, Promotion and Tenure Committee Member, HLC Criterion 3 member, Distance Learning Committee Member and Chair, Academic Technology Committee Member, Maverick Innovation Center Mentor (3 mentors), Coleman Foundation Faculty Fellow (2 fellows), Hiring Committee Chair and Member, CS software and lab management. Regional Director for the International Collegiate Programming Contest, and Denver ACM President, Current Chair of FRC Team 4944 charity, charter member of the board. Mentor FRC teams for 4 years. GJMakerspace founding members. Colorado Space Grand Consortium affiliate directors, Chair Computer Science Scholarship committee. Faculty advisors for ACM (Computer Science) and Cyber Security clubs.

Scholarship: Over two-dozen peer reviewed journals publications, principle software development for MusicEase, ArduinoUnit, java/kiss software projects. Team leads and members for Team Grit DARPA 2015 Robotic Challenge (one of about 20 worldwide), and AgBot challenge (2017). Mentors for dozens of independent study and student showcase projects.

Advising: Faculty advisors for 2 departmental clubs, outreach to high schools for career advising, student advising for our CS and incoming students through programs like Mav Scholars.

4.a.4. Faculty vitas should be included in an Appendix E

4.b. Financial Information (finance and budget)

As part of this narrative, describe any significant increases or decreases in the unit cost of the program during the review period, noting factors that may be affecting costs and the possibility of correcting any deviation within existing resources.

1) Total budget revenues and program expenditures See table 10.

2) Ratio of total expenditures/student credit hours. See table 10.

3) External funding (if applicable): Any external funding the program or its faculty have submitted and received since the last review. What potential opportunities exist for obtaining external funds during the next six years? 30K grant from PNNL for CUDA computational chemistry project, 15k/year ongoing grant for Colorado Space Grant as affiliate program, about 2k/year in travel support through international ACM and other sources for workshop/conference travel.

4.c. Library assessment;

The full assessment is in the Appendix F.

Because of the nature of the field; most resources have moved to online resources since the topics change too quickly to be relevant in any other form. There are some online references which might add value to our students (like Safari Online), but have been expensive.

4.d. Physical facilities;

Our current program has a three labs for general classes (WS 118, WS 120, WS 205). These labs are shared with other programs. A move to the new engineering building was seen by faculty as desirable because this would facilitate collaboration on embedded systems with the Mechanical and eventually Electrical engineering programs. It would also allow us to expand our student lab space and avoid duplications, such as the availability of a fabrication and testing lab instead of our own small version. Given changes in the building design, this option no longer appears viable.

4.e. Instructional technology and equipment;

Significant improvements have been made with the overhead projection systems this year that addressed significant problems with interfacing modern laptops with the old projection systems. Most of our students have laptops and they mostly only have HDMI ports to connect to our campus systems.

An ongoing struggle is the management of attention within the classroom; most students have laptops and mobile devices and we often want them to interact with that technology during a class, but keeping them on task can be problem.

Finally the campus-wide 6 year hardware replacement policy means computer hardware is usually out of date; the IT department does manage these systems and the associated software, but they feel out of date long before they get replaced.

4.f. Efficiencies in the way the program is operated;

We are understaffed and it is unfair to call Lori Payne's full-time load as department chair as an efficiency. When we can find qualified people, the use of part-time non-tenured faculty to teach evening courses is an efficiency that enhances our offerings.

4.g. Other information/data (program's option).

We have been fostering the Maverick Innovation Center through mentoring and co-location of resources. This has been through advising for Karl Castleton, Ram Basnet and Warren MacEvoy, as well as co-location of courses such as Introduction to Robotics and Computer Architecture. From the perspective of our own students, we want to promote a sense of innovation which will hopefully foster new businesses that involve our students.

5. Student Learning Outcomes and Assessments

Student Learning Outcome # 1: Students will write programs in multiple programming languages, and be able to translate concepts between languages. We measure this outcome in CSCI 112 CS 2 Data Structures (D) and CSCI 310 Advanced Programming (A)

CSCI 112: Data Structure (Developing)

This is one of the core courses where students learn fundamental concepts in various data structures and search algorithms. Individual projects are assigned throughout the semester to assess the learning outcomes. As a final project, students implement various common search and sorting algorithms and analyze and compare the running time of those algorithms on an input data of 50K randomly generated integer values.

Results:

4/A: 63%; 3/B: 9%; 2/C: 0%; 1/D: 3%; N/A/F: 25%

A quarter of the students received an F, while close to two-thirds of the class satisfactorily completed the project. Most of the students who received an F didn't turn in the project. It was not clear why students didn't turn in the project, especially when the problem itself was not complicated albeit somewhat longer compared to previous projects they could have done in CS1 and CS2. Students were also given plenty of time -- about a month -- to work on the project. I believe more data needs to be collected to figure out why students received an F and see if it warrants a need for change in the project and/or the assessment method.

CSCI 310 Building a rubric across the 310 courses is a work in progress for measuring this SLO.

Area	Developing	Advanced	Measurement
Language to Machine model	How a program is represented on a computer to run is	The mapping of the application to an effective model of computation.	Ex: write/outline a function (developing) or set of classes (advanced) that compute f(n,p)=sum(k^p,k=1..n). For string

	understood in basic terms. For example, function scope vs global scope variables.	For example, instance vs static scope, and multithreading.	data: write a function that finds the first substring in a string (don't just use a library, implement your own logic). Advanced: cache the results locally so old values are not recomputed in a thread-safe way either through instance storage, thread-local storage, or synchronization locks.
Numeric and Character data	Understanding the basics of data types and using correct data types for a problem.	Understanding the consequences of finite precision in numerical computations and character encodings for international data.	For the above examples, what are good choices made for the types of values. Advanced: what are reasonable limits on n and p for the data types you chose? What are the dangers of multibyte character encodings on your substring search?
Modularization	Elementary modularization of concepts, such as the use of functions to break up the solution to a problem.	Use of objects, namespaces, libraries, and other language-specific features to manage complexity.	Does the student make an appropriate function/method declaration. Advanced, is the caching mechanism logically isolated, and do they take advantage of language features (templates, collections) to efficiently implement them?
Efficiency	Have an idea of the trade-off costs of choosing one data structure or algorithm over another.	Speak of choosing algorithms in terms of big-O complexity.	Time how long it takes to compute some large n and/or long string cases. Advanced: discuss the trade-offs of different choices (hashes vs balanced trees) for the caching technique.
Security	Understand basic security pitfalls in common programming design (bounds checking, data sanitization)	Take a hacker perspective on a design and describe exploitable flaws.	Illustrate what happens when array indices are out of bounds. Advanced: show various code/programs with flaws and discuss flaws in the design.

Architectural Design	Interpret a design to solve a problem in the target language	Use structured techniques to describe a design appropriate to the language (like a UML diagram).	Use a description of a problem to develop a solution. Advanced: use, for example, UML, to illustrate the components and how they are related.
Development Techniques	Use good conventions when writing a program, and basic debugging techniques.	Think about requirements, design, development, and testing and when they are appropriate, and use version control and automated testing.	Ex: give useful names which follow naming conventions appropriate to the language when solving a problem, and know how to debug a program. Advanced: illustrate test-driven design trough multiple check-ins to a version control system of user stories, test driven api design, and then a functional and unit-tested solution to the design requirements.

CSCI 310 Advanced Programming (Advanced)

Results: 4/A: 80%; 3/B: 8%; 2/C: 4%; D/1: 8%; N/A/F: 0%

Results: Most students did well in translating concepts between programming languages when asked to solve the same or similar projects from other programming classes. E.g., the first programming project students typically do in Python course is design and develop a Hangman game which is the final project given to some CS 111 Foundations of Computer Science courses. The project involves fundamental concepts in programming such as input, output, logical statements, iterations and basic arithmetic operations.

Key Findings: Students enjoyed focusing more on the programming syntax, as they are supposed to know the fundamental concepts and principles from other programming languages. Students typically seemed to pick up the syntax much faster and better understand the concepts because of the prior knowledge in other programming languages. They also normally do better on the projects partly because the problems may also have been known from previous classes. Because its an advanced programming class students are given somewhat challenging real-world problems to solve as part of bigger projects. Many students, in faculty evaluations, noted that they liked working on real-world meaningful projects even though they are difficult as opposed to some hypothetical ones.

Conclusions: Students are motivated and enjoy the class if they’re offered fun and simpler programming language such as Python. They tend to be even more motivated when asked to solve some real-world problems. Advanced programming is a fast paced course focused more on program syntax than programming concepts. The results show that majority of the students are able to easily translate concepts learned in C++ to Python and solve the same problem.

Student Learning Outcome # 2: Students will develop the technical specification, and develop, design and test a software solution for a given problem. We measure this outcome in CSCI 490 Software Engineering (Advanced) and CSCI 337 User Interface Design (Developing).

CSCI 337: Teams of students chose projects, for example an interactive campus map application, small construction business application for planning and project management, personnel and billing, and coffee barista application for delivery, and a learning application for elementary children. As a first experience with a team project for most students, all groups succeeded in developing their application. However, only one group included all desired aspects. Overall, every group reached an acceptable level.

Some challenges are understanding the scope of the project, and how much work is involved in the details of the project, or missing entire technical components.

To achieve a higher level of success, we plan on added a phase for understanding the data workflow through an application from the client perspective.

CSCI 490: teams of students choose a project involving an external client to complete during the semester. This is our capstone class.

Results: Most student teams do satisfactorily complete a project. However some teams struggle. The reasons are varied, but a few identified changes are:

Weekly standup meetings describing progress on the project by each team member. These are analogs to daily scrums for full-time software developers. This helps keep students on track and aware of the project timeline.

Version control is introduced early. We are trying to move it very early (introduced in the freshman year), but for now the topic is emphasized and required in the first week. This should help teams keep better management of their project.

Team dynamics are discussed early and revisited during the semester. Working as a team is always challenging and we use the course as a forum for understanding teams and the social structures within them.

Student Learning Outcome # 3: Students will analyze and measure competing hardware and software components and defend a choice for a given situation. This is assessed in CSCI 241 Computer Architecture (Developing) and CSCI 490 Software Engineering (Advanced).

CSCI 241: Students are asked multiple choice questions about hardware choices for the lowest level of computer architecture early in CSCI 241. For example, “What is the propagation delay of a half adder created with only NAND gates compared to the speed of a half adder created with AND, Or, and NOT gates?” Later in CSCI 241 the difference in performance between 16, 32, and 64 bit opcodes and operands for the same high level program statements are used to check students comparative abilities. Questions that compare Acorn RISC Machine (ARM) Architecture performance are compared to Intel x86 performance also serve the purpose of checking students ability to compare different solutions.

Previously this class was taught using a virtual microprocessor tool. While this allowed students to learn the essence of Computer Architecture, it was determined that students had to adapt their understanding to real machines such as ARM that are in cell phones and Intel processors that are in laptop and desktop systems. A change was instituted in 2014 that involves students using lab room Intel processors and network attached ARM processors to perform labs.

CSCI 490: The expectation of this outcome was that students that have hardware choices to make can defend those choices. Since most of the projects have limited hardware (it either is software only or predetermined which hardware components are needed), not all teams had a hardware component to defend. However those that did could and did defend their choices in front of a panel of local software developers (all teams present their work in the spring as a student showcase). No modifications were made in light of the alignment of this outcome with observed results.

Student Learning Outcome # 4A: Students will independently learn and use new technologies. CSCI 370 Computer Security (Developing) and CSCI 490 Software Engineering (Advanced).

CSCI 370: Results: 4/A: 81% A; 3/B: 11%; 2/C: 8%; 1/D: 0%; N/A/F: 0%

Students were able to independently pick the right tools and technologies to assess security vulnerabilities on computers and networks and mitigate the same. Also as the final project, students working in a group of two to three independently picked and researched on a current security topic and presented the findings as the final project.

Key Findings: Though majority of the students demonstrated that they were able to independently learn new techniques and technologies in security, some struggled and didn’t perform as well as expected compared to the students in the previous year (e.g., Fall 2013). Senior students were more motivated and outperformed their junior counterparts. Independently learning new technologies and tools requires that students have their own high quality laptop or desktop system. It was a surprise to find that some students do not own a computer to download and install and try out security tools. As a result it was bit of a hassle for some. To address this problem, alternative projects have been provided and the format of this course has been changed to some extent that it doesn't heavily depend on having a personal system but be able to use the computer lab systems.

Conclusions: It’s important that computer science students have the drive and ability to independently learn and use new technologies in order to succeed and do well in their careers in technology fields. If instructors, mostly teaching advanced and developing level courses, assign more independent real-world projects, we're confident that we'll achieve the outcome.

In CSCI 490, each project is unique and varied enough that inevitable that new ideas will come into play. Every team learned some new technique independently and so this objective was well met and we have no plans to change this structure of the course.

Student Learning Outcome # 4B: Students will work in teams to solve large scale problems. CSCI 337 User Interface Design (Developing) and CSCI 490 Software Engineering (Advanced).

CSCI 337: All students work in a 3-4 person team to complete a project. This is the first major software team project for most students. Many are surprised to find out their peers may not do the work agreed. Some leaders do not delegate effectively. Students do review each other's work at the end of the process. The major component of the project is instructor evaluation on the oral demonstration at the Student Showcase.

One of the challenges is there are peers which are friends but may work poorly together. Distinguishing workmates from friends can be complicated.

A change is to create an official step for teams to choose their team leaders, early enough in the project timeline to be effective but after some team self-assessment can happen.

In CSCI 490, this is the principle semester project. The relatively short time frame and non-full time commitment implies the project is not that large. A challenge for the student teams is to pick a right-sized project that reasonably matches the team skill set and time line.

To address that we discuss the structure of failing – deciding when to cut your losses and move to a new direction. On day 1 this is introduced as the principle problem and encourage students to immediately start sizing up the project they have in mind to limit failures.

6. Future Program Plans

6.a. Vision for program

We want to be the regional resource for computer science and computer science education on the Western Slope. From resources for teachers seeking to enhance their offerings of technically creative uses of computers in K-12 to post-graduate security training.

Addressing the Computer Science For All Whitehouse initiative would be a significant value to the Western Slope; allowing these students be aware of and have basic skills in the field before college.

6.b. Strengths and challenges facing program

Strengths

We have a supportive environment with skilled technically diverse faculty and students willing to challenge themselves.

A flexible program that responds quickly to industry changes. Examples of this are

- Computer Security (including a new certification track)
- Mobile Applications (which didn't exist in 2009)
- Embedded Systems (robotics projects and courses and K-12 outreach)

Our alumni survey gives evidence that our students find full-time employment in the field (90% full time in field), are well compensated (40k-60k average salary), and feel that their degree provided CS background (very well 28%, more than adequate 26%, adequate 38%) and had a good experience while here. The following comment from the student survey illustrates our point:

> *I still maintain contact with some of the Computer Science professors. I received more than an education, it was a start on life. I really enjoyed my years at Mesa and with all the upgrades and new buildings, I wish I could go back and keep learning.*

The Maverick Innovation Center. The program's connection to the Maverick Innovation Center and Mechanical Engineering program provided opportunities that could not have been accomplished alone, such as connections to internships, collaborative startup opportunities, and patent applications for students.

Weaknesses

It is difficult to find qualified faculty, either full time or part-time. This is a very competitive field; it took 3 years to fill two positions. This is a major reason our growth may be limited.

The projected growth from the point of view of students and occupation demand makes our facilities challenging. We share labs with math, developmental math, business and statistics. Our research and experimental labs are tiny. These all limit enrollment and growth.

6.c. Trends in the discipline that could affect future planning for program (if applicable);

Our challenges are to stay abreast of the field, which can completely change in one academic program review cycle, and to make sure we offer a rewarding environment for those who wish to join our program as faculty (it was a struggle to hire Ram and Karl).

There are strong external pressures to increase the number of Computer Science related skills in the U.S.. The BLS put out a 10 year report in 2012 stating the need for these skills in a report entitled "STEM 101: Intro to tomorrows jobs". *Table 2: Selected STEM occupations with many job openings, projected 2012-22* of that document shows nearly one million job openings in occupations that are taught in Computer Science classes. The relevant tables of the BLS report are added as an Appendix C.

In the near term we believe the best strategy to grow our capability will be to find and local qualified adjunct faculty to teach classes we cannot offer with current faculty. Then when there is a consistent number of student-credit hours being taught through that approach, we'll make a case for additional FTE's in the Computer Science program.

One technical question is the use of laptops; the cost of laptops has come to the point they could be required for at least all upper division courses; this may make maintaining a smaller set of machines for the freshman classes with a faster update cycle more feasible.

Computer Security continues to be a strong area of growth in the field; we have addressed that partially through program hires but there are opportunities in professional education and possible master's courses we would like to pursue.

Version control and software testing are two topics which are under-developed in our program, as is indicated by our alumni survey and trends in the field. We are addressing this in our CS0, programming languages, and software engineering classes. That future is now so it is easier to predict.

6.d. How program review process is being used to improve the program's teaching and learning

Since we are a small program, the program review involves the whole department. The discussion about the review and the content you are reading now are the heart-felt opinions of our faculty that are striving to make a better program our of a good one.

Additionally, we have examined our program against other similar campuses, and against the new ACM/IEEE standards. We compare very well as a BS degree, but we are considering adding a new bachelor’s of arts degree in computer science to give a pathway to a computer science degree with fewer math requirements.

The alumni survey brought our attention to some weak areas with respect to unit testing and version control, we are updating our curriculum to address this weakness.

6.e. Recommended program's challenges and potential resources needed to address them

Fitting our moving target into an institutional model can be challenging. For example, requiring homework to be turned into some course shell makes little sense when a CS project may be a course shell. Asking for long-term collection of objectives is also challenging in a subject area that delivers significant changes to society in the course of five years. It also challenges us to accept the expanding roles and interactions of the field with less traditional computer science areas, such as social psychology, mass communications, medicine, and the law. The program needs to remain flexible, experiment and change.

Meshing the goals of careers in computer science conflict with goals of local economic development. Most of our students will be better paid taking a job out of the valley, but we recognize that the information technology needs of the local economy are largely supported by our department, and we hope our students will work to build and grow companies which will have a significant impact on our local economy. Three of our faculty members, Ram Basnet, Karl Castleton, and Warren MacEvoy, work to support entrepreneurship through the Maverick Innovation Center and the Coleman Fellows program, as well as interactions with the local chamber of commerce, LaunchWestCo, and the Grand Junction Business Incubator to foster local economic growth. None of these efforts have been connected to a reduced course load or reduction in other expectations of the department.

Lori Payne has worked hard to support the needs of our program and defended its value and even existence to the university. We should have the staffing to allow for the department chair to have a reasonable amount of time in support of administrative needs and course load.

There is a large justified interest in computer security, and we need to improve our infrastructure so accommodate and expand our computer security offerings. In particular, a class of 40 students may arguably need access to 160 networked virtual machines in a typical attack scenario; we are working to expand our server resources to match this need. If we want to expand consideration to mobile and forensic security, representative hardware and storage facilities would be hard to come by in our current setting.

We have an international reputation in robotics and competed against tier 1 research universities in this area. We would like to spread out from the middle and reach more students in our service area in aliment with the cs-for-all initiatives to expand awareness of the computer science as a technical field in our K-12 schools; paying particular attention to women who are severely underrepresented in our field.

The institutional recognition of the value of information technology through facilities support gives multiple benefits: we need these facilities to effectively grow the program, they represent a commitment by the campus to this obviously central resource to our students that are more and more aware of the value of this kind of degree, and they represent a commitment to the local economy that our campus can meet the technical needs of companies participating in the new economy. The last has been pointed out as a central question for software companies considering where they would like to locate an office.

Curriculum changes to meet the needs of STEM fields and help with program's recruitment and retention efforts

CSCI 110 – Beginning Programming. Program review committee has recommended to change the course title to Problem Solving with Programming. By using Python as a tool, the course contents will be modernized to reflect the changes and requirements in Computer Science industry as a whole. By changing the course, we hope to achieve three major goals: 1) Better prepare and retain Computer Science majors 2) "Catch all type" course to attract undecided students who want to try out Computer Science field, and 3) Serve the needs of STEM fields in their computing needs and help their students better prepare for 21st century work force.

Traditionally, the course was designed for majors outside the scientific disciplines -- especially for Computer Information System majors. The programming language used was Visual Basic. We've seen gradual decline in enrollment in this course over the years partly because, we believe, Computer Information System department now provides its own programming course in C# programming language and their majors are not required to take this course. As a result, we've started teaching the course in Python programming language which will be more appealing to not only our CS majors but to other STEM fields on campus. Python is one of the most popular general purpose programming languages widely used in varieties of disciplines. Plus we believe its simple syntax makes it’s a great choice as the first programming language to learn the basics and fundamentals of programming concepts. Coding is one of fundamental skills for 21st century and we strongly believe that Python will serve well in that purpose.

Moreover, in order to encourage CS students to take this course as the first programming class, 2 credit will be counted towards the 5 credit hours required Advanced programming classes. Important tools such as Git repository and Linux terminals will also be covered in this course to address the current industry needs and trends.

We plan to modify our embedded course offerings by adding an introductory course (CSCI 122, Creating Real World Solutions), that will be appealing to women and men. This will lead to our more advanced CSCI 322 Embedded Systems. CSCI 322 currently depends on CSCI 321, which we will change to CSCI 241 which is more appropriate.

Appendix A

Student Learning Objectives

Appendix A: Student Learning Objectives

Associate Objectives

A1 – Students will have a working knowledge of a general purpose programming language.

A2 – Given a technical specification, students can develop a software solution to a problem.

A3 – Students understand the powers and limitations of basic computer hardware and software.

A4 - Students can communicate technical concepts.

A5 - Students understand the dynamic nature of computer science.

Baccalaureate Objectives

B1 – Students will have a working knowledge of several programming languages, and the ability to translate concepts between languages.

B2 – Given a problem, students can research and develop the technical specification, and develop, design and test a software solution.

B3 – Students can analyze and measure competing hardware and software components and defend a choice for a given situation.

B4 – Students can compare and contrast competing technical methodologies, explaining and defending choices.

B5 – Students will independently learn and use new technologies.

B6 – Students can work in teams to solve large scale problems.

Course Map

COURSE	**TITLE**	**CH**	**GEN ED**	**AS**	**BS**	**MIN-OR**	**O1**	**O2**	**O3**	**O4**	**O5**	**O6**
CSCI 100	Computers In Our Society	3	Y									
CSCI 106	Web Page Design I	3	Y									
CSCI 110 &110L	Beginning Programming	1 - 3										
CSCI 111	CS1: Foundations	4		R	R		AB TH	AB TH	AB TH			

CSCI 112	CS2: Data Structures	4		R	R		A,B T,H	A,B T,H	A,B T,H	A,B T,H	A,B T,H	
CSCI 130	Intro. to Engineering CS	4		O			AB TH	AB TH	AB TH	AB TH		
CSCI 206	Web Page Design II	3		R			A PTH	A PTH	A PRA	A PRA	A PRA	
CSCI 241	Computer Architecture	4		R	R	O			AB THR	AB TH	AB TH	
CSCI 250	CS3: Intro to Algorithms	3		R	R	R	AB TH	AB TH	AB TH	AB TH	AB TH	
CSCI 306	Web Page Design III	3			O	O	B PTH	B PTH	B PRA	B PRA	B PRA	B PRA
CSCI 310	Advanced Programming	1 - 3		O	R	O	AB H	AB H	AB H		AB H	
CSCI 322	Embedded Systems	3			O	O		B HTP	B HTP	B RA	B HTRAP	
CSCI 330	Programming Languages	3			R		B PTHRA		B PTHRA	B PTHRA	B P	B PGA
CSCI 333	Unix	3			O	O	B PHRA	B PHRA	B P	B P	B P	
CSCI 337	User Interface Design	3			O	O	B HP	B P	B PT	B PT	B PR	B PGA
CSCI 345	Video Game Design	3			O		B GPH	B GPH	B GPH	B GPHA	B GPH	B GPHA
CSCI 370	Computer Security	3			O				B TH	B TH	B TH	
CSCI 375	Object Oriented Programming	3			O	O	B THP	B THP	B THP	B THP	B THP	
CSCI 380	Operations Research	3			O			B TH	B TH			
CSCI 445	Computer Graphics	3			O		B THP	B THP	B THP	B THP	B THP	
CSCI 450	Compiler Design	3			O		B HPR	B HPR	B HPR		B PR	B PR
CSCI 460	Database Design	3			O	O	B THP	B THP	B THP	B THP	B THP	

CSCI 470	Operating Systems	3			R			B THRP	B THRP	B THRP	B RP	
CSCI 480	Theory of Algorithms	3			O		B TH	B TH	B TH	B TH	B TH	
CSCI 482	Theory of Computation	3			O				B THRA	B THRA	B THRA	
CSCI 484	Computer Networks	3			R			B THRP	B THRP	B THRP		
CSCI 486	Artificial Intelligence	3			O		B THR	B THR	B THR	B THR	B THR	
CSCI 490	Software Engineering	3			R			B GPRA	B GPRA	B GPRA	B GPRA	B GPRA
MATH 151	Calculus I	5			R							
MATH 152	Calculus II	5			R							
MATH 369	Discrete Mathematics	3			R							
STAT 200	Probability & Statistics	3			R							
CISB 205	Advanced Business Software	3	O									

Key

R	Required
O	Option in program choice list
A#	Associate objective (see below)
B#	Baccalauriate objective (see below)
G	Group/team work
P	Project (part or full semester)
T	Test
H	Homework/program
R	Reading/research
A	Analyze/present

Appendix B

Alumni Survey Results:

Combined 2013-2016

Appendix B: Alumni Survey Results - Combined 2013-2016

Alumni Survey Results for Computer Science Graduates

(n = 18)

Overall, how satisfied are you with your undergraduate education?

	#	%
Very Satisfied	3	16.7%
Generally satisfied	14	77.8%
Ambivalent	1	5.6%
Generally Dissatisfied	0	0.0%
Very Dissatisfied	0	0.0%

While an undergraduate, about how often did you have conversations with faculty outside of class?

	#	%
Never	0	0.0%
Rarely (1-2 times per semester)	3	16.7%
Occasionally (3-5 times per semester)	3	16.7%
Often (once every two weeks)	6	33.3%
Very Often (at least once a week)	6	33.3%

Would you encourage a current high school senior to attend CMU?

	#	%
Definitely Would	7	38.9%
Probably Would	10	55.6%
Maybe	1	5.6%
Probably Would Not	0	0.0%
Definitely Would Not	0	0.0%

In what year did you graduate from the major/certificate you chose above?

	#	%
2015	2	11.1%
2014	8	44.4%
2013	4	22.2%
2012	2	11.1%
2011	0	0.0%
2010	2	11.1%
Other	0	0.0%

How would you rate the overall quality of your education within that degree/certificate program?

	#	%
Very High	2	11.1%
High	10	55.6%
Average	6	33.3%
Low	0	0.0%
Very Low	0	0.0%

	Very Well		More than Adequately		Adequately		Less Than Adequately		Very Poorly	
	#	%	#	%	#	%	#	%	#	%
Write Programs in multiple programming languages, and be able to translate concepts between languages.	3	16.7%	5	27.8%	9	50.0%	1	5.6%	0	0.0%
Develop the technical specification, and develop, design, and be able to translate concepts between languages.	2	11.1%	9	50.0%	4	22.2%	2	11.1%	1	5.6%
Analyze and measure competing hardware and software components and defend a choice for a give situation	3	16.7%	3	16.7%	9	50.0%	3	16.7%	0	0.0%
Independently learn and use new technologies	5	27.8%	3	16.7%	7	38.9%	3	16.7%	0	0.0%
Work in teams to solve large scale problems	3	16.7%	6	33.3%	4	22.2%	4	22.2%	1	5.6%

Student Comments

- As a graduate student, all are relevant. As a software engineer, Software Engineering, Web Design III, Object Oriented Programming, Database Design, Unix
- I moved into the Operations side (hardware,networking,servers,troubleshooting,deployments of software,RHEL upgrades,automation of all-the-things,monitoring,scripting,etc) of things after I graduated, but I am part of a DevOps team that works hand in hand with people who only write code, in this case Ruby on Rails, to develop cloud-based solutions to some of the most critical hospital systems in the US. Having a Computer Science degree and having written code myself for years allows me to work in these tight-knit teams very well, as I can converse with coders on

their level, unlike many of my co workers that got degrees from tech schools in networking, computer engineering, etc.

- I graduated with a statistics degree, and the analytical material was very helpful.
- N/A
- Unix and complier design were actually very helpful because of the diversity of education they encouraged. I also write quite a bit of embedded C so learning everything in C++ turned out to be quite useful.
- Data structures
- Object Oriented Design, Unit Testing, Algorithms
- Being able to translate my knowledge between languages.
- Debugging code across multiple operating systems. C based languages, understanding the difference between compiled and interpreted languages.
- AI, Compilers, Algorithms.
- Currentry still doing appliance repair. Java has been most useful learning android development
- The ability to translate between languages really quickly and efficiently. I'm often finding myself moving between Android development, server development, sql, etc.
- Networking

Baccalaureate Degree Student Learning Outcomes

Based on what you know now, how well do you think your undergraduate experience prepared you to:

	Very Well		More than Adequately		Adequately		Less Than Adequately		Very Poorly	
	#	%	#	%	#	%	#	%	#	%
Construct a summative project, paper or practiced-based performance that draws on current research, scholarship and/or techniques, and specialized knowledge in the discipline (Applied Learning/ Specialized Knowledge)	2	11.1%	3	16.7%	11	61.1%	2	11.1%	0	0.0%
Analyze data critically, reason logically, and apply quantitative analysis methods correctly to develop appropriate conclusions (Intellectual Skills: Quantitative Fluency)	2	11.1%	10	55.6%	4	22.2%	2	11.1%	0	0.0%

Make and defend assertions about a specialized topic in an extended well-organized document and an oral presentation that is appropriate to the discipline (Intellectual Skills: Communication Fluency)	1	5.6%	9	50.0%	3	16.7%	5	27.8%	0	0.0%
Identify assumptions, evaluate hypotheses or alternative views, articulate implications and formulate conclusions (Intellectual Skills: Critical Thinking)	5	27.8%	6	33.3%	6	33.3%	1	5.6%	0	0.0%

In what type of organization is your principal employment? Mark the one best answer.

Self-employed in own business or professional non-group practice	1
Private for-profit corporation/company/group/group-practice	12
Higher education (public or private)	0
Elementary or secondary education (public or private)	0
International organization in the US	1
International organization outside of the US	0
US Military	1
Federal Government (except military)	0
State and local government, institution, or agency (except education)	0
Private non-profit organization (except education and international organizations)	0
Other	0

Which of the following best describes your current position?

	#	%
Entry Level	7	41.2%
Mid-Level	6	35.3%
Senior Level	3	17.6%
Executive Level (except for chief executive)	0	0.0%
Chief Executive (CEO, COO, CFO, GM or principal in a business of other organization)	1	5.9%

How many years have you been in your current job type?

	#	%
Less than 3 years	12	70.6%
3-5 years	3	17.6%
6-9 years	2	11.8%
10 or more years	0	0.0%

Is your current position related to your undergraduate field(s) of study?

	#	%
Yes, related to major(s)	14	82.4%
No, not related	3	17.6%

How well did CMU prepare you for your current career?

	#	%
Very Well	2	11.8%
More than Adequately	4	23.5%
Adequately	7	41.2%
Less Than Adequately	1	5.9%
Very Poorly	0	0.0%
NA	3	17.6%

What is your approximate annual gross income (before taxes)?

	#	%
Under $20,000	1	6.7%
$20,000 - $29,999	0	0.0%
$30,000 - $39,999	0	0.0%
$40,000 - $49,999	0	0.0%
$50,000 - $59,999	2	13.3%
$60,000 - $74,999	5	33.3%
$75,000 - $99,999	4	26.7%
$100,000 - $149,999	3	20.0%
$150,000 - $249,999	0	0.0%
$250,000 - $499,999	0	0.0%
Over $500,000	0	0.0%

Comments about your work experience that will help improve CMU:

- The environment of the CS program was incredible. It was rather laid back and relationships with the professor made the learning environment very welcoming. I don't think CMU has the most rigorous academics for CS, yet I don't believe this is a negative. I personally would have been discouraged early on if the program was much more rigorous. I really enjoyed learning CS comfortably, and I feel that the quality of education is self driven, but available for students who truly wish to learn.
- Unit tests are very important at work and these were only briefly covered in my CMU education
- meh
- There is not much demand for statisticians in the Grand Valley. So when the jobs come around nearly all ask for 3+ years of work experience. Can't build work experience for jobs if the jobs aren't around.
- I still maintain contact with some of the Computer Science professors. I received more than an education, it was a start on life. I really enjoyed my years at Mesa and with all the upgrades and new buildings, I wish I could go back and keep learning.
- More focus on testing, both unit and integration. More team work and working with version control software. More OOP design concepts as well and functional programming concepts. Mutability vs immutability, concurrency and parallelism. Different types of abstractions between OOP and functional programming paradigms.

Why are you not currently working for pay? (Please mark all that apply)

	# of times checked
I chose not to enter the workforce at this time.	0
It has been difficult to find a position in my field.	1
It has been difficult to find a position paying an appropriate salary.	0
I am raising a family.	0
I am currently a student.	1
I am doing volunteer work.	0
I am retired.	0
Other	0

Other Responses:

If you have comments about previous employment, work experience, or job hunting that will help improve CMU, please leave them here:

- Try focusing on local businesses rather than almost forcing students to leave town just to find employment. There are enough possibilities within the Grand Junction area, provided more active communication and better relationships between them and CMU can be established. School District 51 should also be considered as a main focus for starting a career in the Computer-Science/IT industry, both as teachers and as members of the IT department, which is currently very understaffed to my knowledge.

Education since College

Have you enrolled in a graduate, professional, or other degree/certificate program since graduating from CMU?

	#	%
Yes	4	22.2%
No	10	55.6%
No, but I plan to enroll in the next two years.	4	22.2%

Are you enrolled in this program now?

	#	%
Yes, I am a full-time student	2	50.0%
Yes, I am a part-time student	1	25.0%
No	1	25.0%

How long after you graduated from the degree/certificate program this survey pertains to did you start this program?

	#	%
Immediately (following fall or spring)	2	50.0%
1 Year later	1	25.0%
2-3 years later	1	25.0%
4-6 years later	0	0.0%
NA	0	0.0%

Altogether, how many years have/did you attend(ed) further schooling? Mark the best answer.

	#	%
None	1	25.0%
1 - 2 years	2	50.0%
3 - 4 years	1	25.0%
5 - 6 years	0	0.0%
NA	0	0.0%

How well did CMU prepare you for this educational program?

	#	%
Very Well	1	25.0%
More than Adequately	1	25.0%
Adequately	2	50.0%
Less Than Adequately	0	0.0%

Very Poorly	0	0.0%
NA	0	0.0%

What level of education are/were you pursing?

	#	%
Certificate	0	0.0%
Associate	0	0.0%
Baccalaureate	1	25.0%
Post-Bacc Certificate	0	0.0%
Master's	3	75.0%
J.D.	0	0.0%
Doctoral	0	0.0%
Other	0	0.0%

In which field and program are/were you studying?

- Computer Science
- Mathematics, Secondary Education Concentration
- Masters in Cyber Security
- Computer Science

What is the name of the College/University where you attend(ed)?

- Colorado School of Mines
- Colorado Mesa University
- Bellevue University
- University of Colorado Boulder

Did you complete this program?

	#	%
Yes	1	25.0%
No	1	25.0%
In the process of finishing	2	50.0%

Other comments about furthering your education:

- I am graduating this May. Also, I was sought out to be an instructor for an introductory Programming Concepts class.

Suggestions for improving the degree/certificate program:

- Keep the department a welcoming environment. It makes a world of difference.

- The low level focus early on is good, but the program was lacking in high level areas such as threading and threading models. I also recommend teaching unit tests until students can write unit tests and do Test Driven Development in their sleep. Teaching how to design solutions before implementing them would also be good (e.g. requiring students to learn UML)
- nothing comes to mind.
- For the statistics department, multiple statistical programming needs to be included and taught, most stats-related jobs don't utilize R programming.

Additional Comments:

- A special thanks to Lori Payne, Warren MacEvoy, and Carl Castleton.

Demographic Questions

What is your gender?

	#	%
Male	16	88.9%
Female	0	0.0%
Prefer not to respond	2	11.1%

What is your ethnicity?

	#	%
American Indian or Alaskan Native	0	0.0%
Asian	0	0.0%
Black or African American	0	0.0%
Hispanic of any race	0	0.0%
Native Hawaiian or Pacific Islander	0	0.0%
White	17	94.4%
Two or more races	0	0.0%
Race and ethnicity unknown	0	0.0%
Non-Resident Alien (of any race or ethnicity)	0	0.0%
Prefer not to respond	1	5.6%

What is your current age?

	#	%
Under 21	0	0.0%
21-24	4	22.2%
25-34	14	77.8%
35-44	0	0.0%

45-54	0	0.0%
55 or older	0	0.0%
Prefer not to respond	0	0.0%

Do you live in the state of Colorado?

	#	%
Yes	8	44.4%
No	10	55.6%

If yes, do you live in Western Colorado?

	#	%
Yes	5	27.8%
No	13	72.2%

Appendix C

BLS Job Projections

Appendix C: BLS Job Projections

Table 2: Selected STEM occupations with many job openings, projected 2012–22

Occupation	Job openings, projected 2012–22	Employment		Median annual wage, May 2013	Typical entry-level education[1]
		2012	Projected 2022		
Software developers, applications	218,500	613,000	752,900	$92,660	Bachelor's degree
Computer systems analysts	209,600	520,600	648,400	81,190	Bachelor's degree
Computer user support specialists[2]	196,900	547,700	658,500	46,620	Some college, no degree
Software developers, systems software	134,700	405,000	487,800	101,410	Bachelor's degree
Civil engineers	120,100	272,900	326,600	80,770	Bachelor's degree
Computer programmers	118,100	343,700	372,100	76,140	Bachelor's degree
Sales representatives, wholesale and manufacturing, technical and scientific products[2]	111,800	382,300	419,500	74,520	Bachelor's degree
Network and computer systems administrators	100,500	366,400	409,400	74,000	Bachelor's degree
Mechanical engineers	99,700	258,100	269,700	82,100	Bachelor's degree
Computer and information systems managers[3]	97,100	332,700	383,600	123,950	Bachelor's degree
Industrial engineers	75,400	223,300	233,400	80,300	Bachelor's degree
Architectural and engineering managers[3]	60,600	193,800	206,900	128,170	Bachelor's degree
Web developers	50,700	141,400	169,900	63,160	Associate's degree
Electrical engineers	44,100	166,100	174,000	89,180	Bachelor's degree
Computer network architects[3]	43,500	143,400	164,300	95,380	Bachelor's degree

[1] Unless otherwise specified, occupations typically require neither work experience in a related occupation nor on-the-job training to obtain competency.

[2] In addition to the education specified, this occupation typically requires moderate-term on-the-job training for workers to obtain competency.

[3] In addition to the education specified, this occupation typically requires 5 years or more of work experience in a related occupation.

Source: U.S. Bureau of Labor Statistics, Employment Projections program (employment, projections, and education data) and Occupational Employment Statistics survey (wage data).

Table 3: Selected STEM occupations with fast employment growth, projected 2012–22

Occupation	Employment growth, projected 2012–22 (percent)	Employment		Median annual wage, May 2013	Typical entry-level education[1]
		2012	Projected 2022		
Information security analysts[2]	37%	75,100	102,500	$88,590	Bachelor's degree
Operations research analysts	27	73,200	92,700	74,630	Bachelor's degree
Statisticians	27	27,600	34,900	79,290	Master's degree
Biomedical engineers	27	19,400	24,600	88,670	Bachelor's degree
Actuaries[3]	26	24,300	30,600	94,340	Bachelor's degree
Petroleum engineers	26	38,500	48,400	132,320	Bachelor's degree
Computer systems analysts	25	520,600	648,400	81,190	Bachelor's degree
Software developers, applications	23	613,000	752,900	92,660	Bachelor's degree
Mathematicians	23	3,500	4,300	102,440	Master's degree
Software developers, systems software	20	405,000	487,800	101,410	Bachelor's degree
Computer user support specialists[4]	20	547,700	658,500	46,620	Some college, no degree
Web developers	20	141,400	169,900	63,160	Associate's degree
Civil engineers	20	272,900	326,600	80,770	Bachelor's degree
Biological science teachers, postsecondary	20	61,400	73,400	75,740	Doctoral or professional degree
Environmental science and protection technicians, including health	19	32,800	38,900	41,700	Associate's degree

[1] Unless otherwise specified, occupations typically require neither work experience in a related occupation nor on-the-job training to obtain competency.

[2] In addition to the education specified, this occupation typically requires less than 5 years of work experience in a related occupation.

[3] In addition to the education specified, this occupation typically requires long-term on-the-job training for workers to obtain competency.

[4] In addition to the education specified, this occupation typically requires moderate-term on-the-job training for workers to obtain competency.

Source: U.S. Bureau of Labor Statistics, Employment Projections program (employment, projections, and education data) and Occupational Employment Statistics survey (wage data).

Appendix D

CS Retention

Appendix D: CS Retention

Computer Science Major Profile, Fall 2011-2015

Degree	Major	2011	2012	2013	2014	2015
		#	#	#	#	#
AS	Liberal Arts-Computer Science	38	36	31	40	31
PB	Computer Science Prov Bacc		4	11	12	18
BS	Computer Science	109	119	119	124	149

Degree	Major	2011		2012		2013		2014		2015	
		Avg Index	# Scores	Avg Index	# Scores	Avg Index	# Scores	Avg Index	# Scores	Avg Index	# Scores
AS	Liberal Arts-Computer Science	84	13	84	17	87	13	81	23	81	18
PB	Computer Science Prov Bacc			79	3	84	11	85	12	86	18
BS	Computer Science	105	80	103	89	106	94	106	102	106	120

Degree	Characteristic	2011		2012		2013		2014		2015	
		#	%	#	%	#	%	#	%	#	%
	Class Standing										
AS	FR	23	61%	19	53%	20	65%	22	55%	18	58%
	SO	15	39%	17	47%	11	35%	18	45%	13	42%

PB	FR			2	50%	10	91%	12	100%	18	100%
	SO			2	50%	1	9%				
BS	FR	32	29%	36	30%	33	28%	41	33%	65	44%
	SO	21	19%	17	14%	30	25%	27	22%	24	16%
	JR	26	24%	24	20%	16	13%	23	19%	28	19%
	SR	30	28%	42	35%	40	34%	33	27%	32	21%
	Registration Status										
AS	First-time	8	21%	6	17%	8	26%	10	25%	5	16%
	Transfer	2	5%	5	14%	2	6%	5	13%	2	6%
	Readmit	5	13%	2	6%	1	3%	5	13%	4	13%
	Continuing	23	61%	23	64%	20	65%	20	50%	20	65%
PB	First-time			2	50%	10	91%	7	58%	15	83%
	Readmit			1	25%			1	8%		
	Continuing			1	25%	1	9%	4	33%	3	17%
BS	First-time	19	17%	23	19%	19	16%	28	23%	43	29%
	Transfer	13	12%	5	4%	6	5%	10	8%	9	6%
	Readmit	7	6%	4	3%	5	4%	6	5%	5	3%
	Continuing	70	64%	86	72%	88	74%	79	64%	92	62%
	High School			1	1%	1	1%	1	1%		
	Full/Part-time										
AS	Full-time	30	79%	29	81%	25	81%	30	75%	27	87%
	Part-time	8	21%	7	19%	6	19%	10	25%	4	13%
PB	Full-time			4	100%	11	100%	11	92%	17	94%
	Part-time							1	8%	1	6%
BS	Full-time	99	91%	98	82%	102	86%	107	86%	136	91%
	Part-time	10	9%	21	18%	17	14%	17	14%	13	9%

	Pell-Eligible										
AS	No	11	29%	12	33%	17	55%	17	43%	13	42%
	Yes	27	71%	24	67%	14	45%	23	58%	18	58%
PB	No					1	9%	2	17%	10	56%
	Yes			4	100%	10	91%	10	83%	8	44%
BS	No	58	53%	69	58%	66	55%	66	53%	83	56%
	Yes	51	47%	50	42%	53	45%	58	47%	66	44%

Degree	Characteristic	2011		2012		2013		2014		2015	
		#	%	#	%	#	%	#	%	#	%
	Race/Ethnicity										
AS	African American	1	3%	3	8%	2	6%	1	3%	1	3%
	Hispanic	5	13%	4	11%	3	10%	6	15%	3	10%
	Native American	1	3%	2	6%	1	3%				
	Multi-Racial	1	3%	1	3%	1	3%	1	3%	1	3%
	White	26	68%	23	64%	22	71%	29	73%	24	77%
	Not Reported	4	11%	3	8%	2	6%	3	8%	2	6%
PB	African American					1	9%				
	Hispanic			2	50%	2	18%	3	25%	6	33%
	White			1	25%	8	73%	9	75%	11	61%
	Not Reported			1	25%					1	6%
BS	Asian	2	2%	1	1%	4	3%	4	3%	7	5%
	Pacific Islander					1	1%	1	1%		
	African American	5	5%	3	3%	4	3%	1	1%	3	2%
	Hispanic	9	8%	13	11%	14	12%	14	11%	19	13%

	Multi-Racial	1	1%	3	3%	5	4%	5	4%	4	3%
	White	82	75%	90	76%	86	72%	94	76%	110	74%
	Non-Resident Alien	3	3%			1	1%	1	1%	1	1%
	Not Reported	7	6%	9	8%	4	3%	4	3%	5	3%
	Gender										
AS	Male	35	92%	32	89%	29	94%	38	95%	29	94%
	Female	3	8%	4	11%	2	6%	2	5%	2	6%
PB	Male			4	100%	10	91%	12	100%	16	89%
	Female					1	9%			2	11%
BS	Male	97	89%	106	89%	109	92%	112	90%	134	90%
	Female	12	11%	13	11%	10	8%	12	10%	15	10%
	Age										
AS	Traditional aged	17	45%	15	42%	18	58%	22	55%	17	55%
	Non-traditional	21	55%	21	58%	13	42%	18	45%	14	45%
PB	Traditional aged			3	75%	11	100%	11	92%	18	100%
	Non-traditional			1	25%			1	8%		
BS	Traditional aged	76	70%	77	65%	81	68%	88	71%	113	76%
	Non-traditional	33	30%	42	35%	38	32%	36	29%	36	24%
	Region										
AS	14-County	33	87%	30	83%	21	68%	28	70%	25	81%
	Front Range	4	11%	4	11%	7	23%	10	25%	4	13%
	Other Colorado	1	3%					1	3%	1	3%
	WUE			1	3%	1	3%				
	Mtns/Plains			1	3%	1	3%				
	Other					1	3%	1	3%	1	3%

PB	14-County			3	75%	6	55%	6	50%	11	61%
	Front Range					3	27%	2	17%	5	28%
	Other Colorado			1	25%	1	9%	2	17%	1	6%
	WUE							1	8%	1	6%
	Mtns/Plains					1	9%	1	8%		
BS	14-County	77	71%	74	62%	72	61%	81	65%	99	66%
	Front Range	24	22%	32	27%	33	28%	27	22%	32	21%
	Other Colorado	3	3%	6	5%	7	6%	5	4%	3	2%
	WUE	2	2%	6	5%	4	3%	6	5%	7	5%
	Mtns/Plains			1	1%	2	2%	3	2%		
	Other	3	3%			1	1%	2	2%	8	5%

		2011		2012		2013		2014		2015	
Degree	Characteristic	#	%	#	%	#	%	#	%	#	%
	Housing										
AS	Off-campus	34	89%	31	86%	25	81%	32	80%	27	87%
	On-campus	4	11%	5	14%	6	19%	8	20%	4	13%
PB	Off-campus			2	50%	6	55%	6	50%	9	50%
	On-campus			2	50%	5	45%	6	50%	9	50%
BS	Off-campus	81	74%	93	78%	91	76%	97	78%	117	79%
	On-campus	28	26%	26	22%	28	24%	27	22%	32	21%

Retention of first-time, full-time, bachelor's-seeking freshmen

	2011		2012		2013		2014	
	#	%	#	%	#	%	#	%
Computer Science majors in cohort	19		21		19		26	

Retained in same major	8	42%	11	52%	8	42%	15	58%
Retained other major	3	16%	2	10%	1	5%	3	12%
Subtotal Retained	11	58%	13	62%	9	47%	18	69%
Campus-wide Retention		65%		65%		66%		70%

6-year Graduation Rate of Fall 2009 First-time, Full-time, Bachelor's Seeking Freshmen

	#	%
Computer Science majors in cohort	12	
Did not graduate	7	58%
BS - Computer Science	5	42%
Total graduating in 6 years	5	42%
Campus-wide graduation rate		37%

Degrees Awarded, Academic Year

Degree	Major	2010-11	2011-12	2012-13	2013-14	2014-15
AS	Liberal Arts-Computer Science		3	2	3	3
BS	Computer Science	7	15	16	23	21
Total		7	18	18	26	24

Appendix E

CVs

Appendix E: CVs

In order of hire:

- Lori Payne,
- Gary Rader,
- Warren MacEvoy,
- Ram Basnet,
- Karl Castleton, and
- John Dobbs.

Name:

Lori K. Payne

Start Year: 1986

Program:
Computer Science

Department:
Computer Sciences, Mathematics, and Statistics

COLORADO MESA UNIVERSITY

Full-time Faculty Vita

Faculty Rank

- (●) Professor
- ( ) Assistant Professor
- ( ) Associate Professor
- ( ) Instructor

Highest Degree

PhD	University of Northern Colorado	Education Technology - Interactive Technologies	1996

Education: (List all degrees beginning with most recent-include post docs and external certificates)
Ph.D., Educational Technology - Interactive Technologies, University of Northern Colorado,1996.
AWU Faculty Fellowship, 1991.
M.S., Groundwater Hydrology, New Mexico Institute of Mining & Technology, 1984.
B.A., Mathematics & Computer Science, Mesa College, 1978.
A.S., Mathematics, Mesa College, 1976.

Teaching 2003-Present:
Courses Taught
CSCI 111 CS1: Foundations of Computer Science

CSCI 112 CS2: Data Structures

CSCI 250 CS3: Intro. To Algorithms

CSCI 310 Advanced Programming: Visual Basic

CSCI 310 Advanced Programming: Delphi

CSCI 330 Programming Languages

CSCI 337 User Interface Design

CSCI 396 Independent Study

CSCI 460 Database Design

CSCI 480 Theory of Algorithms

MATH 113 College Algebra

Math 110 College Mathematics

MATH 361 Numerical Analysis

STAT 200 Probability & Statistics

Stat 214 Business Statistics

Name:

Gary M Rader

Start Year: 1995

Program:
Computer Science

Department:
Computer Sciences, Mathematics, and Statistics

COLORADO MESA UNIVERSITY

Full-time Faculty Vita

Faculty Rank
(•) Professor ( ) Assistant Professor
( ) Associate Professor ( ) Instructor

Highest Degree

PhD	University of Pennsylvania	Computer and Information Science	1976

Education: (List all degrees beginning with most recent-include post docs and external certificates)
M. B. A., University of Phoenix, 1984.

Ph.D., Computer and Information Science, University of Pennsylvania, 1976.
Dissertation: Computer Generation of Simple Stories and Their Variations.
Committee: Saul Gorn, Rob Ross, Norm Badler, Aravind Joshi, and Ruzena Bajcsy.

M.A., Computer and Information Science, University of Pennsylvania, 1973.
Thesis: An Algorithm for the Automatic Composition of Simple Forms of Music Based on a Variation of Formal Grammars.
Supervisors: Saul Gorn and Constance Vauclain.

B. A., Mathematics, University of Pennsylvania, 1968.

Teaching 2003-Present:
Courses Taught
Computers and Society
Introduction to Artificial Intelligence
Programming Languages
C++ Programming I and II
Web Page Design I and II
Data Structures
Introduction to Algorithms
Advanced Algorithms

Innovative Materials/Activities
creation of real applications

Scholarship and Creative Work, 2003-Present:
Creative Work Related to Discipline

Publications
MusicEase Traditional Songbook, 2003, (2012 Mac App store).
MusicEase (notation software), 1988-Present.
Christian Virtual Hymnal, 2005-Present.
Christian Virtual Hymnal Lite, 2011 (Mac App store).
Christmas Carols Songbook, 2011 (Mac App store).
Messiah MusicEase, 2011 (Mac App store).

Name:

Warren D MacEvoy

Start Year: 1996

Program:
Computer Science

Department:
Computer Sciences, Mathematics, and Statistics

COLORADO MESA UNIVERSITY

Full-time Faculty Vita

Faculty Rank
◉ Professor ○ Assistant Professor
○ Associate Professor ○ Instructor

Highest Degree

PhD University of Arizona Applied Mathematics 1995

Education: (List all degrees beginning with most recent-include post docs and external certificates)

NSF Industrial and Applied Mathematics Postdoc New York University and Schlumberger-Doll Research, 1995

Ph.D., Applied Mathematics, University of Arizona, 1995

M.S., Applied Mathematics, University of Arizona, 1991

B.S., Computer Science and Mathematics, Mesa College, CO, 1989

Teaching 2003-Present:
Courses Taught
CSCI 106 Web Page Desgin 1
CSCI 130 Introduction to Engineering Computer Science
CSCI 306 Web Page Design 3
CSCI 310 Advanced Programming: C#
CSCI 310 Advanced Programming: Java
CSCI 321 Assembly Language Programming
CSCI 333 Unix Operating Systems
CSCI 322 Embedded Systems
CSCI 375 Object Oriented Programming
CSCI 405 Mobile Applications
CSCI 445 Computer Graphics
CSCI 450 Compiler Structure
CSCI 460 Database Design
CSCI 470 Operating Systems Design
CSCI 484 Computer Networks
CSCI 486 Artificial Intelligence
PHYS 301 Introduction to Space Science

<u>Evidence of Continuous Improvement</u>
International Collegiate Programming Contest Regional Director's Meeting 1999-Present
SEED Security Education Workshop, Syracuse 2015
ArduinoUnit project lead developer 2013-present
Java Kiss project lead developer 2016

<u>Innovative Materials/Activities</u>
Robotics Projects for Math Extravaganza
Student Server Research Lab

<u>Supervision of Student Research/Project(s)</u>

Ram B. Basnet, PhD, CEH
Assistant Professor of Computer Science
Colorado Mesa University
1100 North Avenue, Grand Junction, CO 81501
Office: (970) 248-1682
Email: rbasnet@coloradomesa.edu
Web: **http://org.coloradomesa.edu/~rbasnet**

EDUCATION

New Mexico Tech, Socorro, New Mexico

Ph.D. in Computer Science, 2010 – 12

Dissertation: Detecting Phishing Attacks: a Comprehensive Approach

Research Advisor: Prof. Andrew H. Sung

M.S. in Computer Science, summa kum laude, 2005 – 07

Colorado Mesa University, Grand Junction, Colorado

B.S. in Computer Science and Minor in Mathematics, summa kum laude, 2001–04

INDUSTRY CERTIFICATE

EC-Council, http://www.eccouncil.org

Certified Ethical Hacker (CEH), 2013 - Present

License SR9203326

ACADEMIC POSITION

Colorado Mesa University, Grand Junction, Colorado

Assistant Professor of Computer Science, 2013 – Present

- Created Cyber Security Professional Certificate Program, Dec. 2015
- Created Cyber Security Club, Aug. 2015

SCHOLARSHIPS, HONORS AND AWARDS

Colorado Mesa University, Grand Junction, Colorado

- Faculty Professional Travel Grant, 2013-14, 2015-16
- Faculty Professional Development Grant, 2015-16
- The President's List, 2001 - 04
- Rim Rock International Scholarship, 2001 – 04

- Wayne N. Aspinall Foundation Science Scholarship, 2003 – 04
- Mesa State College Foundation - Bacon Family Scholarship, 2002 – 04
- Member, Kappa Mu Epsilon
- Member, Upsilon Pi Epsilon

New Mexico Tech, Socorro, New Mexico

- The Chancellor's List, 2005 – 06

Little Angels' College, Kathmandu, Nepal

- Outstanding Student of the Batch, 1997 – 99
- Best All Rounder, 1997

RESEARCH ACTIVITIES

JOURNAL AND CONFERENCE PUBLICATIONS

- T. Doleck, **R. Basnet**, E. Poitras, and S. Lajoie, "Mining Learner-system interaction data: implications for modeling learner behaviors and improving overlay models." Journal of Computers in Education, Springer, Aug. 2015.
- T. Doleck, **R. Basnet**, E. Poitras and S. Lajoie, "Towards examining learner behaviors in a medical intelligent tutoring system: A Hidden Markov Model approach." In Proc. Of the 2015 IEEE International Conference on Advance Computing Conference (IACC 2015), June 2015, Banglore, India.
  R. B. Basnet and T. Doleck, "Towards Developing a Tool to Detect Phishing URLs: A Machine Learning Approach." *In Proceedings of International Conference on Computational Intelligence and Communication Technology (ICICT)*, India: IEEE. 2015.
- B. M. Ribeiro, A. H. Sung, D. Suryakumar, and **R. B. Basnet**, "The Critical Feature Dimension and Critical Sampling Problems." *In Proceedings of the 4th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2015)*, Lisbon, Portugal, January 2015.
- T. Doleck, **R. B. Basnet**, E. Poitras, and S. Lajoie, "Exploring the Link Between Initial and Final Diagnosis in a Medical Intelligent Tutoring System." *In Proceedings of IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)*, India: IEEE. 2014.
- T. Doleck, **R. B. Basnet,** E. Poitras, and S. Lajoie, "BioWorldParser: A Suite of Parsers for Leveraging Educational Data Mining Techniques." *In Proceedings of IEEE International Conference on MOOC, Innovation and Technology in Education (MITE)*, India: IEEE, 2014.
- T. Doleck, **R. B. Basnet**, E. Poitras and S. Lajoie, "Augmenting the Novice-Expert Overlay Model in an Intelligent Tutoring System: Using Confidence-Weighted Linear Classifiers." *In Proceedings of IEEE International Conference on Computational Intelligence & Computing Research (ICCIC),* Tamil Nadu, India, 2014.

- **R. B. Basnet** and A. H. Sung, "Learning to Detect Phishing Webpages." *JISIS - Journal of Internet Services and Information Security, pp. 21-39, Vol. 3, No. 4, August, 2014.*
- **R. B. Basnet** and A. H. Sung, and Q. Liu, "Learning to Detect Phishing URLs" *IJRET - International Journal of Research in Engineering and Technology, Vol. 3, Issue 6, June 2014.*
- **R. B. Basnet** (2012), "Detecting Phishing Attacks: a Comprehensive Approach," *Ph.D. Dissertation*, New Mexico Tech, 2012.
- **R. B. Basnet** and Andrew H. Sung (2012), "Mining Web to Detect Phishing URLs," *11th International Conference on Machine Learning and Application (ICMLA), 2012.*
- **R. B. Basnet**, A. H. Sung, and Q. Liu (2012), "Feature Selection for Improved Phishing Detection," *Data Mining and Computational Intelligence for Digital Forensics and Information Assurance (IEA/AIE 2012)*.
- **R. B. Basnet**, A. H. Sung, and Q. Liu (2011), "Rule-Based Phishing Attack Detection," *Proc. of International Conference on Security and Management*, pp. 624-630.
- **R. B. Basnet** and A. H. Sung (2010), "Classifying Phishing Emails Using Confidence-Weighted Linear Classifiers," *Proc. of International Conference on Information Security and Artificial Intelligence,* pp. 108-112.
- G. J. Torres, **R. B. Basnet**, A. H. Sung, and S. Mukkamala (2009), "A Similarity Measure for Clustering and its Applications," *International Journal of Electrical and Electronic Engineering*, pp.164-170.
- **R. B. Basnet** and S. Mukkamala (2009), "Event Detection and Localization Using Sensor Networks," *Proc. of International Conference on Wireless Networks*, pp. 103-108.
- M. K. Shankarapani, **R. B. Basnet**, S. Mukkamala, A. H. Sung, and B. M. Ribeiro (2008), "Translation Based Arabic Text Categorization," *Proc. of 2nd International Conference on Information Systems Technology and Management*.
- **R. B. Basnet**, S. Mukkamala, and A. H. Sung (2008), "Detection of Phishing Attacks: A Machine Learning Approach," *Studies in Fuzziness and Soft Computing*, Springer, pp. 373-383.
- S. Mukkamala, K. Yendrapalli, **R. B. Basnet**, and A. H. Sung (2007), "Detecting Coordinated Distributed Multiple Attacks," *Proc. of 21st International Conference on Advanced Information Networking and Applications*, pp. 557-562.
- S. Mukkamala, K. Yendrapalli, **R. B. Basnet**, M. K. Shankarpani, and A. H. Sung (2007), "Detection of Virtual Environments and Low Interaction Honeypots," *Proc. of Information Assurance and Security Workshop, IEEE SMC*, pp. 92-98.
- **R. B. Basnet**, K. Yandrapali, T. Dolek, and S. Mukkamala (2007), "Knowledge Based Discovery of Phishing Attacks," IICAI, 2007.
- S. Mukkamala, A. H. Sung, K. Yendrapalli, and **R. B. Basnet** (2007), "Biased Support Vector Machine and Kernel Methods for Tumor Classification," Rocky Mountain Bioinformatics Conference, Aspen, CO, 2007.
- **R. B. Basnet,** G. Torres, A. H. Sung, S. Mukkamala, B. Ribeiro (2008), "Translation Based Foreign Language Text Categorization," unpublished (https://cs.nmt.edu/~rbasnet/research/TBTC.pdf).

INVITED TALKS AND PRESENTATIONS

- *Ethical Hacking*, a keynote presentation, February 2015

17th Annual Math Extravaganza, Colorado Mesa University

- *An Anatomy of a Cyber Attack: How script kiddies can own a network*, October 2014
  Mathematics Colloquium, Colorado Mesa University
- *Secure Digital World – Be a Hero*, April 2014
  R-5 High School, Grand Junction, Colorado.
- *Rule-Based Phishing Attack Detection*, November 2013
  Mathematics Colloquium, Colorado Mesa University

PROFESSIONAL ACTIVITIES

- Reviewer: Elsevier - Computers and Security Journal, 2011 – Present
- Coleman Fellow of Entrepreneurship Education, 2014 – Present
- Program committee: Data Mining and Computational Intelligence for Digital Forensics and Information Assurance (*IEA/AIE*), 2012
- Member of Team Grit Robotics (formerly Team Mojavaton), a Colorado team for DARPA Robotics Challenge, 2013 - Present
- A strong supporter and promoter of gocode.colorado.gov, 2015 - Present

RESEARCH EXPERIENCE

New Mexico Tech, Socorro, New Mexico
Graduate Research Associate, 5/2010 – 12/2011

- Supported research and development.
- Contributed in research proposal and grand writing.
- Published research and findings in peer-reviewed conferences and journals.

Research Scientist II, 1/2008 – 4/2010

- Researcher and lead developer on CACTUS (Computational Analysis of Cyber Terrorism against the United States), $1,000,000 project funded by federal grant; focused on research and development in text and Web mining; collaborated and integrated software modules in Natural Language Processing developed by Computing Research Laboratory (crl.nmsu.edu) at New Mexico State University.
- Directed innovative design and development of a scalable information gathering, retrieval, and visualization framework.
- Supervised graduate students and interns on various projects such as network security and forensics, machine learning, data mining, client and compliance management systems, application security, etc.
- Identified and demonstrated XSS and SQL Injection vulnerabilities on New Mexico Legislator website when it was compromised during the 2008 session; reviewed and fixed the source codes written in ASP and JSP and brought the site back online in 4 days.
- Designed and developed an application called Tweezer to gather, analyze (sentiment analysis), and present results on tweets from Twitter using Django (MVC) framework.
- Developed teaching materials (http://code.google.com/p/owasp-top-10/) and provided hands-on trainings on Top 25 Most Dangerous Programming Errors and OWASP Top 10 Errors to developers from New Mexico State departments and BudgeText Corp. (www.budgetext.com).

- Performed information security posture assessments for various New Mexico state agencies such as taxation and revenue, judicial information systems, legislative council service, corrections department, etc. and private businesses.

Graduate Research Assistant, 8/2005 – 12/2007

- Performed research and development in network and web application security.
- As a founding member of the “Strike Team”, participated in network security posture assessment of various government agencies and private firms.
- Analyzed and used several commercial and open-source network security and digital forensic software tools.

INDUSTRY EXPERIENCE

Sage Technology Partners, Inc., Albuquerque, New Mexico
Chief Technology Officer, 4/2012 – 6/2013

- Lead R&D and consulting efforts in cyber security services especially in network and web application security.
- Performed network and application security assessment and penetration testing for various private and government entities.
- Developed custom applications for clients.
- Lead the effort in developing various in-house software tools and applications.
- Provided cyber security awareness training to clients.
- Evaluated and implemented various security and IT products for managed services.
- Revamped corporate web presence; developed marketing strategy using social media.

Flow-Data, Inc., Grand Junction, Colorado
Software Developer, 9/2004 – 8/2005

- Designed and developed desktop and web applications in VB6 and ASP.NET.
- Designed and maintained company’s website.

Freelance Web Developer: 5/2004 – 8/2006.

- Developed and maintained e-commerce website for Black Canyon Foods, LLC using ASP.NET.

Killian & Davis, PC, (formerly Killian, Guthro, and Jensen, PC), Grand Junction, Colorado
Runner, Network & System Administrator Support, 7/2004 – 7/2005

- Provided IT support, maintained systems and network, designed and maintained databases and website, designed ads, etc.

Mesa State College, Grand Junction, Colorado
Computer Lab Technician, 8/2001 – 5/2004

- Provided computer lab help desk support; fixed and setup computers, printers, and network devices, etc.

SERVICES

Colorado Mesa University, Grand Junction, Colorado

- Faculty Advisor, Cyber Security Club, 8/2015 - Present
- Faculty Advisor, Upsilon Pi Epsilon, 8/2013 – Present
- Maverick Innovation Center Committee, Fall 2014 - Present
- Scholarship Committee Chair, CSMS, Fall 2015 – Present
- Scholarship Committee, CSMS, Fall 2013 – Present
- Search Committee: Computer Science Tenure-Track Position, 8/2013 – 8/2014
- Faculty Facilitator, Student Showcase, 2014
- Commencement Ceremony Volunteer, May, Dec. 2014, Dec. 2015

Community

- Organizer, Mesa County Middle and High School Science Fair, March 2015
- Promoter, Go Code Colorado App Challenge, 2014 - Present
- Mentor, Grand Junction High School Cyber Tigers Club, 8/2013 - 8/2014

STUDENT ADVISING AND MENTORING

Colorado Mesa University, Grand Junction, Colorado

- Tenzin Doleck, PhD student, McGill University, 2014 - Present
- Spring 2016 – 38 advisees
- Fall 2015 – 46 advisees
- Spring 2015 – 34 advisees
- Morgan Creekmore, Deploying 10K small VSs in CS Server, 2014
- Paul Jensen, Hac5 Pineapple WiFi, 2014
- John Madison, Piratebox, 2014
- Fall 2014 – 34 advisees
- Spring 2014 - 24 advisees
- Fall 2013 – 10 advisees

PROJECTS

- Go Code Colorado App Challenge Finalist, 2015
- EVENT (Enterprise Vulnerability Exploration & Network Topology) - consolidates network vulnerability assessment reports from industry best scanners and generates prioritized custom reports for all levels of IT staff. Provides a unified GUI framework for popular network security scanners (Nmap, Amap, and Nessus Client). The framework also contains exploit module.
- MAKE2 (Media Analysis Knowledge Extraction & Exploration) - a compressive digital forensic toolkit. Major features include: disk imaging, file system exploration, MAC time visualization, full-text indexing and searching, email graph visualization, image search and visualization, hex viewer, etc.
- ARIN WhoisRWS API Wrapper - http://code.google.com/p/pywhoisapi/.
- Python API for longurl.org - http://code.google.com/p/pylongurl/.
- Hands-on learning materials on OWASP Top 10 and Top 25 Most Dangerous Programming Errors for asp.net developers - http://code.google.com/p/owasp-top-10/.

TEACHING EXPERIENCE

Colorado Mesa University, Grand Junction, Colorado

Undergraduate Level

- Web Page Design 1 (3 cr) - FA13, SP14, FA14, SP15, FA15
- Foundations of Computer Science (4 credits) - SP14, FA14, SP15, FA15
- Computer Security (3 cr) – FA13, FA14, FA15
- Topics: Cybersecurity (3 cr) – SP14
- Cybersecurity (3 cr) – SP15, SP16
- Topics: Net/App Security (3 cr) – FA15
- Data Structure (3 cr) – SP16
- Advanced Programming using Python (2 cr) – FA13, FA14, SP16
- Beginning Programming using Python (4 cr) – SP15, SP16

New Mexico Tech, Socorro, New Mexico

Workshop Training

- Hands-on Secure Web Application Design and Development, New Mexico State software developers, Santa Fe, New Mexico, 2009
- Hands-on Secure Web Application Design and Development, BudgeText Corp, Little Rock, Arkansas, 2009

WORKSHOP, CONFERNECE AND TRAINING ATTENDED

- Faculty Professional Development Program, Colorado Mesa University, January 2016
- Cybersecurity Summer Research and Training for College Faculty, New York University, New York City, July-Aug 2015.
- Coleman Foundation Entrepreneurship Fellow Summit, Chicago, August 2014.
- Faculty Development Workshop, Colorado Mesa University, August 2014.
- General Education Workshop, January 2014.
- NDSS (Network and Distributed System Security) Symposium, San Diego, February 2014.

TECHNICAL SKILLS

- Python, C++, C#, JAVA, VB, C, Matlab, Bash Script, Perl
- ASP.NET, PHP, JSP, ASP, HTML, XML, CSS, JavaScript, Silverlight, MVC
- MySQL, MSSQL, Sqlite, MS Access, OOP/OOD (UML, Design Patterns), GUI Design, Web Services, etc.
- Nessus, LANGuard, SPI Dynamics, Retina, XScan, Nmap, Amap, Metasploit, Canvas, Qualys, Acunetix, WebInspect, Nikto, Wireshark, pyflag, dd, The Sleuth Kit, COFEE, EnCase, FTK, Kali Linux, etc.
- Mac, Linux, Windows

PROFESIONAL AFFILIATIONS

- Member, IEEE
- Member, ACM

Full-time Faculty Vita

Name:

Karl J Castleton

Start Year: 2001

Program:
Computer Science

Department:
Computer Sciences, Mathematics, and Statistics

Faculty Rank
○ Professor ○ Assistant Professor
◉ Associate Professor ○ Instructor

Highest Degree

MS Washington State University Computer Science 1997

Education: (List all degrees beginning with most recent-include post docs and external certificates)

MS CS Washington State University 1997

BS CS and Mathematics Colorado Mesa University (formerly Mesa State College) 1992

AAS Electronics Technology Mesa State College 1987

Teaching 2003-Present:
CSCI 100 Computer in our Society
CSCI 110 Beginning Programing Visual Basic
CSCI 106 Web Page Design 1
CSCI 196 Introduction to Robotics
CSCI 241 Computer Architecture and Assembly Language
CSCI 206 Web Page Design 2
CSCI 310 Advanced Programming: Python
CSCI 333 Unix Operating Systems
CSCI 345 Video Game Design
CSCI 396 Advanced Robotics
CSCI 484 Computer Networks

Evidence of Continuous Improvement
2015 Intel's Hardware Accelerator Program as part of a project for interactive clothing.
2015 U.C. Berkeley's Startup Program as part of project in interactive clothing.
Invited to DARPA Robotics Challenge Finals in Spring 2015. Only 24 teams from around world invited to this event.

Innovative Materials/Activities
Made substantial use of 3D printing to create a more cost effective robot for the DARPA Robotics Challenge.

Supervision of Student Research/Project(s)
Coprincipal Investigator NASA Space Grant at Colorado Mesa University. Oversee about 10 students every semester working on robotics. Many of these students participated in the DARPA Robotics Challenge from 2013 to 2015

Scholarship and Creative Work, 2003-Present:
An integrated environmental modeling framework for performing Quantitative Microbial Risk Assessments Whelan, G; Kim, K; Pelton, MA; Soller, JA; Castleton, KJ; Molina, M; Pachepsky, Y; Zepp, R, ENVIRONMENTAL MODELLING & SOFTWARE, 55 77-91; 10.1016/j.envsoft.2013.12.013 MAY 2014

Design of a component-based integrated environmental modeling framework Whelan, G; Kim, K; Pelton, MA; Castleton, KJ; Laniak, GF; Wolfe, K; Parmar, R; Babendreier, J; Galvin, M ENVIRONMENTAL MODELLING & SOFTWARE, 55 1-24; 10.1016/j.envsoft.2014.01.016 MAY 2014

Whelan G, ME Tryby, MA Pelton, JA Soller, and KJ Castleton. 2010. "Using an Integrated, Multi-disciplinary Framework to Support Quantitative Microbial Risk Assessments." In Proceedings of the International Congress on Environmental

Appendix F

Library Assessment

Appendix F: Library Assessment

Library Program Assessment
John U. Tomlinson Library
Colorado Mesa University
Date of Assessment: September 2016

Purpose of Assessment: Program Review

Program under review: Computer Science

Program Level: Bachelors

Liaison: Jay Ballenberger

I. Collection Assessment:

Reference Support: A search of the Tomlinson Library catalog for reference items appropriate to the Computer Science program resulted in numerous relevant reference collection results, including these published since 2005:

(Titles are **bold**; e-reference titles in **green**)

- Alhajj, R., & Rokne, J. (2014). ***Encyclopedia of social network analysis and mining***. New York: Springer
- Benthem, J. V., & Meulen, A. T. (2011). *Handbook of logic and language.* Amsterdam; Boston: Elsevier
- Butterfield, A. P., & Ngondi, G. E. (2016). ***A Dictionary of Computer Science***. Oxford University Press.
- Daintith, J., & Wright, E. (2008). *A dictionary of computing.* Oxford University Press, 2008
- Date, C. J. (2008). *The relational database dictionary.* Berkeley, CA: Apress; New York: Distributed to the Book trade in the United States by Springer-Verlag
- Downing, D., & Covington, M. (2012). ***Dictionary of computer and internet terms***. Barron's Educational Series, 11th edition.
- Engquist, B. (2015). *Encyclopedia of applied and computational mathematics.* Heidelberg: Springer
- Henderson, H. (2009). ***Encyclopedia of computer science and technology***. New York, NY: Facts On File
- Holt, T. J., & Schell, B. H. (2013). ***Hackers and hacking: a reference handbook***. Santa Barbara, California: ABC-CLIO
- Koranne, S. (2011). *Handbook of open source tools.* New York: Springer
- Ozsu, M. T., & Liu, L. (2009). *Encyclopedia of database systems.* New York; London: Springer
- Tilborg, H. v., & Jajodia, S. (2011). *Encyclopedia of cryptography and security.* New York: Springer
- Whitaker, J. C. (2005). ***The electronics handbook***. Boca Raton, FL: CRC Press, 2005.

Analysis: Overall, the reference collection offers sufficient support for this program. Print sources for dictionaries in particular should be acquired as relevant materials become available.

A. Monographic Resources: The following table displays representative results relevant to the Computer Science program within the Tomlinson Library catalog. It shows the number of titles for print items and the number of e-Book items found, as well as those published within the last five years for both categories.

Search tactic:	Print Books	e-Books
"computer science" (Keyword)	**361** (21 since 2011)	**13,502** (6,350 since 2011)
"artificial intelligence" (Keyword)	**162** (15 since 2011)	**3,874** (2,128 since 2011)
"information technology" (Keyword)	**491** (7 since 2011)	**6,032** (3,817 since 2011)
"computer networks" (Keyword)	**231** (22 since 2011)	**5,352** (3,075 since 2011)
"software engineering" (Keyword)	**71** (14 since 2011)	**4,122** (2,652 since 2011)
"computer programming" (Keyword)	**331** (25 since 2011)	**4,137** (1,972 since 2011)
"Computer architecture" (Keyword)	**68** (12 since 2011)	**2,171** (1,248 since 2011)
"Computer hardware" (Keyword)	**39** (9 since 2011)	**1,742** (1,052 since 2011)

Monographic titles of interest include:
(Titles are **bold**; e-book titles in **green**)

- Guttag, J. (2013**).** ***Introduction to computation and programming using Python****.* Spring 2013 edition. Cambridge, Massachusetts: The MIT Press
- Hey, A. J. G., & Pápay, G. (2015). ***The computing universe: a journey through a revolution****.* New York, NY, USA: Cambridge University Press.
- Lewis, P. R., Platzner, M., Rinner, B., Torresen, J., & Yao, X. (2016). *Self-aware computing systems: an engineering approach*. Switzerland: Springer
- Loveland, D. W., Hodel, R. E., & Sterrett, S. G. (2014). ***Three views of logic: mathematics, philosophy, and computer science****.* Princeton: Princeton University Press.
- Neittaanmaki, P., Repin, S. I., & Tuovinen, T. (2016). *Mathematical modeling and optimization of complex structures*. Cham: Springer
- Pollack, E. (2016). *Dynamic SQL: applications, performance, and security*. United States: Apress
- Torbert, S. (2016). *Applied computer science*. Switzerland: Springer, 2016.

Analysis: In general, our monograph collection serves the program faculty and students well, as it is quite strong and current. The liaison will continue to acquire relevant titles in consultation with program faculty.

> **NOTE:** Books and other physical items outside of our collection may be requested from partner institutions via **Interlibrary loan** with 3-5 day average delivery time. These items are free to CMU students, staff, and faculty and provide valuable support for those engaged in research and scholarship. Our interlibrary loan service can also aid the Library in assessing frequently requested items for appropriate additions to our collection.

B. Periodicals: A search of the CMU Tomlinson Library collection of print and electronic periodical subscriptions highlights a number of relevant titles. A selection of useful titles we subscribe to for this program include:

- *ACM Transactions on Computer Systems* (Business Source Complete, 02/01/2009 to current)
- *ACM Transactions on Programming Languages & Systems* (In print: 07/01/1986 to 09/01/1994, and Business Source Complete: 01/01/2009 to current)
- *Computer Languages* (Science Direct, 04/01/1995 to 12/31/2001)
- *International Journal of Foundations of Computer Science* (Academic Search Complete, 03/01/99 to 09/09/2015)
- *International Journal of Information Security* (Academic Search Complete, 08/01/2001 to 09/09/2015)
- *Journal of the Brazilian Computer Society* (DOAJ, 01/01/2014 to current)
- *Journal of Computer Security* (Academic Search Complete, 03/01/1996 to 12/09/2015)
- *Journal of Logic and Algebraic Programming* (Science Direct, 01/01/2001 to current)
- *Journal of the ACM* (Business Source Complete, 03/01/2004 to current)
- *Proceedings of the American Mathematical Society* (JSTOR, 02/01/1950 to 12/01/2010)
- ***Proceedings of the London Mathematical Society*** **(Academic Search Complete,** 01/01/1997 to 09/09/2015)
- ***SIAM Journal on Scientific Computing*** **(Academic Search Complete,** 01/01/1997 to 07/01/2011)
- ***Theoretical Computer Science*** **(Science Direct,** 01/01/1975 to current)

Analysis: Periodical holdings for the Computer Science program are strong. Also, Tomlinson Library subscribes to hundreds more journals that are useful to students and faculty within the program, for assigned readings and the fulfillment of assignments through the use of appropriate sources.

> **NOTE:** Periodicals to which we do not subscribe and/or articles for which we do not offer full-text access can be delivered via **Interlibrary loan** to supplement our collections, with an average turn-around time of just 10.5 hours for articles. Such fast request-to-delivery times greatly aid students in their information needs being addressed and processed as close to the time of need as possible.

C. Electronic Resources:

Business Source Complete [searched: “computer science” OR “information technology” OR “computer security”] – over 335,000 search results, nearly 94,000 of which were published since 2011 and are available full-text; over 48,000 are scholarly, peer-reviewed sources.
Science Direct Freedom Collection [searched: “computer science” OR “information technology”] – over 314,000 search results, nearly 119,000 of which were published since 2011 and are available full-text.
Wiley Online Library [searched: “computer science” OR “information technology” OR “computer security”] – over 860,000 search results, nearly 363,000 of which were published since 2011 and are available full-text.
JSTOR [searched: “computer science” OR “information technology” OR “computer security”] – over 13,000 search results, nearly 1,000 of which were published since 2011 and are available full-text.

Analysis: These databases offer high-quality, relevant resources to support the program, and provide students and faculty with reputable and appropriate readings for the execution of assignments and other course work/course development needs.

II. Evaluation of the Total Collection:

a. Strengths: The Library’s reference collection supports the program well, both with print and electronic resources; the Library will continue efforts to collect current, highly relevant titles, particularly in consultation with program faculty. With regard to monographic, periodical, and database content, the collection is sufficient to meet a wide variety of student needs within the program, both in terms of the breadth of topics covered and access to print and electronic items.

b. Weaknesses: There are few weaknesses within the collection, as the coverage of this subject at Tomlinson Library is generally strong. Two new print reference titles have been acquired with the intention of providing faculty and students with more current and reputable computer science dictionaries. Overall, consistent efforts will continue to be made to collect relevant content addressing shifts/growth in the field, changes to program curriculum, and access to electronic items, both to support students who may not routinely come to campus but who have need of library resources, and particularly for studies within such a fast growing field as Computer Science.

III. Recommendations:

Based on the findings in this assessment, the library liaison to the program recommends general maintenance to the collection – weeding no-longer-relevant titles, acquiring titles to support new/reconfigured program courses/needs, and working with department faculty to acquire titles of high interest, including but not limited to reference titles.

IV. Library Research Support Statement:

The Library can offer research support for specific courses through **instructional sessions** (how to access and use our research tools, how to evaluate information and sources, how to use information ethically, etc.) as well as through an **embedded librarian** in the D2L environments of course sections. The Library can also create **research / course guides** for course sections and entire subjects, tailoring links, tools, e-reserve access, and more to the students and the topic of that section, course, or subject area.

Program faculty and instructors can make use of any/all of these options, and the Library is interested in investigating customized possibilities with them.

Library Director:

Date: October 10, 2016

Appendix G

Majors and Minors for Computer Science

Appendix G: Majors and Minors for Computer Science

Table 1 - Majors and Minors for Computer Science

		2011-12		2012-13		2013-14		2014-15		2015-16	
Degree	Major	1st	All	1st	All	1st	All	1st	All	1st	All
AS	Liberal Arts - Computer Science	5256	54	44	56	39	49	46	50	32	39
PB	Computer Science Prov. Bacc.	2	2	5	5	12	12	12	12	14	14
BS	Computer Science	124	7200	129	145	130	139	139	150	175	186
		5382	7256	178	206	181	200	197	212	221	239

Minor	11/12	12-13	13-14	14-15	15-16
Computer Science	6	5	5	8	7

Appendix H

Degrees Awarded

Appendix H: Degrees Awarded

Table 2 - Degrees Awarded Computer Science

Degree	Major	2011-12		2012-13		2013-14		2014-15		2015-16	
		1st	All	1st	All	1st	All	1st	All	1st	All
AS	Liberal Arts - Computer Science	3	3	2	2	3	3	3	3	2	2
BS	Computer Science	14	15	15	16	23	23	20	21	20	21
		17	18	17	18	26	26	23	24	22	23

Minor	11/12	12-13	13-14	14-15	15-16
Computer Science	1	1	0	0	1

Appendix I

Registrations and Student Credit Hours by Student Level

Appendix I: Registrations and Student Credit Hours by Student Level

Table 3 - Registrations and Student Credit Hours by Student Level

		2011-12		2012-13		2013-14		2014-15		2015-16	
Subject	Student Level	1st	All	1st	All	1st	All	1st	All	1st	All
CSCI	Freshmen	145	460	149	493	108	366	131	440	111	378
	Sophomore	235	750	225	722	236	764	223	723	272	859
	Junior	155	492	142	440	167	531	150	478	197	603
	Senior	333	1007	382	1128	345	1017	327	988	356	1025
	PBL	1	4	1	3	2	6	1	3	0	0
	Graduate	1	3	0	0	0	0	1	3	0	0
	Graduate Non-Deg	0	0	0	0	0	0	0	0	3	11
	Non Degree	0	0	4	12	3	10	2	6	2	7
		870	2716	903	2798	861	2694	835	2641	941	2883

Appendix J

Registrations and Student Credit Hours by Course Level

Appendix J: Registrations and Student Credit Hours by Course Level

Table 4 - Registrations and Student Credit Hours by Course Level

		2011-12		2012-13		2013-14		2014-15		2015-16	
Subject	Course Level	1st	All	1st	All	1st	All	1st	All	1st	All
CSCI	100	502	1623	478	1556	454	1501	479	1564	495	1575
	200	96	319	93	309	74	243	98	329	103	347
	300	148	402	216	585	210	581	178	508	221	595
	400	124	372	116	348	123	369	80	240	122	366
		870	2716	903	2798	861	2694	835	2641	941	2883

Appendix K

Registrations and Student Credit Hours by Majors/Non-Majors

Appendix K: Registrations and Student Credit Hours by Majors/Non-Majors

Table 5 - Registrations and Student Credit Hours by Majors/Non-Majors

		2015-16	
Subject	Major?	Enrolled	SCH
CSCI	Computer Science Majors	606	1855
	Other CSMS Majors	41	137
	Non Majors	294	891
		941	2883

Appendix L

Registrations and Credit Hours for Essential Learning Courses

Appendix L: Registrations and Credit Hours for Essential Learning Courses

Table 6 - Registrations and Credit Hours for Essential Learning Courses

Subject	Title	Cr Hr	2015-16 Sections	Enrolled	SCH
CSCI 100	Computers In Our Society	3	2	56	168
			2	56	168

Appendix M

Registrations and Student Credit Hours by Course Level

Appendix M: Registrations and Student Credit Hours by Course Level

Table 7 - Registrations and Student Credit Hours by Course Level

		2011-12		2012-13		2013-14		2014-15		2015-16	
Subject	Course Level	1st	All	1st	All	1st	All	1st	All	1st	All
CSCI	100	502	1623	478	1556	454	1501	479	1564	495	1575
	200	96	319	93	309	74	243	98	329	103	347
	300	148	402	216	585	210	581	178	508	221	595
	400	124	372	116	348	123	369	80	240	122	366
		870	2716	903	2798	861	2694	835	2641	941	2883

Appendix N

Computer Science Credit Hour Production by Course

Appendix N: Computer Science Credit Hour Production by Course

Table 8 Computer Science Credit Hour Production by Course

	2011-12			2012-13			2013-14			2014-15			2015-16			Head Count*
Faculty Type	CCH	SCH	%SCH	CCH	SCH	%SCH	CCH	SCH	%SCH	CCH	SCH	%SCH	CCH	SCH	%SCH	
T/TT	102	2144	79%	121	2239	80%	93	1641	61%	129	2203	83%	129	2474	86%	5
FT NonTT	3	66	2%	6	111	4%	9	171	6%	9	195	7%	9	198	7%	
Admin	7200		0%			0%	2	42	2%			0%			0%	
PT	25	506	19%	27	448	16%	48	840	31%	13	243	9%	11	211	7%	2
	7330	2716		154	2798		152	2694		151	2641		149	2883		

Appendix O

Computer Science Ratio of Full-time Equivalent Students (FTES) to Full-time Equivalent Faculty (FTEF)

Appendix O: Computer Science Ratio of Full-time Equivalent Students (FTES) to Full-time Equivalent Faculty (FTEF)

Table 9 Computer Science Ratio of Full-time Equivalent Students (FTES) to Full-time Equivalent Faculty (FTEF)

2011-12			2012-13			2013-14			2014-15			2015-16		
FTES	FTEF	FTEF: FTES	FTES	FTEF	FTEF: FTES	FTES	FTEF	FTEF: FTES	FTES	FTEF	FTEF: FTES	FTES	FTEF	FTEF: FTES
90.5	5.4	16.8	93.3	6.4	14.6	89.8	6.3	14.3	88.0	6.3	14.0	96.1	6.2	15.5

Appendix P

Computer Science Expenditures/Student Credit Hours

Appendix P: Computer Science Expenditures/Student Credit Hours

Table 10 Computer Science Expenditures/Student Credit Hours

	2010-11	2011-12	2012-13	2013-14	2014-15
Expenditures	439097	464517	531370	515634	522196
SCH	2482	2716	2798	2694	2641
Cost/SCH	$176.91	$171.03	$189.91	$191.40	$197.73

Appendix Q

CSCI Courses

Appendix Q: CSCI Courses

Table 11 Computer Science Courses

CSCI		
100	Computers In Our Society	3
104	5256	1
106	Web Page Design I	3
110	Beginning Programming: V Basic	7200
110L	Beg. Programming Laboratory	1
111	CS1: Found of Computer Science	4
112	CS2: Data Structures	4
130	Intro to Engineering Comp Sci	3
196	Topics	1-3
206	Web Page Design II	3
241	Computer Arch/Assembly Lang	4
250	CS3: Intro to Algorithms	3
296	Topics	2
305	Tech for Math Educators	3
306	Web Page Design III	3
310	Advanced Programming	2-3
330	Programming Languages	3
333	UNIX Operating Systems	3
337	User Interface Design	3
345	Video Game Design	3
370	Computer Security	3
375	Object Oriented Programming	3
395	Independent Study	1-3
396	Topics	1-3
420	Cyber Security	3
445	Computer Graphics	3
450	Compiler Structure	3
460	Database Design	3
470	Operating Systems Design	3
480	Theory of Algorithms	3
484	Computer Networks	3
486	Artificial Intelligence	3
490	Software Engineering	3

Appendix R

Computer Science Credit Hour Production by Course

Appendix R: Computer Science Credit Hour Production by Course

Table 12 Computer Science Credit Hour Production by Course

	2011-12			2012-13			2013-14			2014-15			2015-16		
Course	#Sec	Enrld	SCH	#Sec	Enrld	SCH	#Sec	Enrld	SCH	#Sec	Enrld	SCH	#Sec	Enrld	SCH
CSCI															
100	2	55	165	2	44	132	2	47	141	2	37	111	2	56	168
104	7200									1	9	9	1	11	11
106	9	211	633	11	196	588	9	151	453	8	170	510	6	121	363
110	2	34	102	2	24	72	1	13	39	2	23	69	2	27	81
110L	2	12	12	2	10	10	1	5	5	2	10	10	2	14	14
111	3	96	384	4	105	420	4	118	472	4	104	416	4	126	504
112	2	45	180	2	41	164	2	51	204	2	61	244	2	54	216
130	3	49	147	3	51	153	3	57	171	3	65	195	3	66	198
196				2	26	66	2	12	16				2	29	43
206	2	32	96	2	30	90	2	26	78	2	25	75	2	21	63
241	1	31	124	1	31	124	2	21	84	2	35	140	2	38	152
250	2	33	99	2	31	93	2	27	81	2	38	114	2	44	132
296				1	1	2									
305							1	5	15	1	6	18	1	13	39
306	1	21	63	1	13	39	1	17	51	1	13	39	1	13	39
310	4	61	141	3	66	154	3	68	162	4	50	126	5	80	176
330	1	30	90	1	24	72	1	21	63	1	17	51	1	29	87
333	1	19	57	1	23	69	1	18	54	1	17	51	1	24	72
337				1	26	78				1	19	57	1	16	48
345	1	16	48	1	22	66	1	19	57	1	18	54	1	9	27
370				1	17	51	1	25	75	1	15	45	1	19	57
375							1	14	42	1	12	36	1	9	27
395	1	1	3	5	6	7	1	1	2	2	2	4			
396							3	22	60	1	9	27			
420													1	12	36
445				1	16	48				1	10	30			
450	1	8	24				1	8	24				1	4	12
460	1	20	60	1	16	48	1	17	51	1	15	45	1	20	60
470	1	22	66	2	23	69	1	28	84	1	19	57	1	24	72
480	1	14	42	1	12	36	1	12	36	1	7	21	1	10	30
484	1	25	75	1	26	78	1	21	63	1	21	63	1	22	66
486	1	16	48				1	8	24				1	9	27
490	1	19	57	1	23	69	1	29	87	1	8	24	2	21	63
Total	44	870	2716	55	903	2798	51	861	2694	51	835	2641	52	941	2883

Appendix S

Budget Data for Computer Science

Appendix S: Budget Data for Computer Science

Table 13a. 5 Year Budget Data for 1460 - Computer Science

		FY11 2010-2011		FY12 2011-12		FY13 2012-13		FY14 2013-14		FY15 2014-15	
Expenditures	Banner Acct#	$	FTE	$	FTE	$	FTE	$	FTE	$	FTE
Contractual - Reg FT	5256	346,383	5.00	333,803	4.50	376,466	5.00	279,135	3.50	393,901	5.00
CN Annual Leave Payments	5386					551					
Contractual - Temp FT	5296	7,200	0.40	34,923	0.83	37,328	1.17	76,000	1.00	6,395	0.23
CNTPT WITH STUDENT FTE	5366	1,360				2,550		28,451	1.00	3,950	
CN DCPP TIAA CREF	5416	13,949		11,279		15,584		16,399		16,955	
CN DCPP FIDELITY	5426			0		0				0	
CN DCPP VALIC	5436	6,721		6,818		7,322		7,444		7,636	
CN FICA-MEDICARE CONTRIBUTION	5446	4,892		5,119		5,820		5,302		5,445	
CN OTHER RETIREMENT PLAN	5456							0		0	
CN PERA	5466	14,958		14,532		19,403		14,790		14,842	
CN PERA-AED	5468	3,055		5,254		6,191		5,281		5,898	
CN PERA-SAED	5469	1,939		4,243		5,349		4,780		5,545	
CN DENTAL INSURANCE	5506	2,897		3,022		2,917		2,882		3,114	
CN HEALTH INSURANCE	5516	25,609		37,192		41,531		49,258		51,305	
CN HSA MATCHING	5520	1,975									
CN LIFE INSURANCE	5526	1,486		1,439		1,406		1,244		1,238	
CN DISABILITY	5536	922		812		915		817		836	
Federal Workstudy Match	5606							92			
Student Assistants	5706	0									

Professional Fees - External	0	0		7		
Employee Travel - In State	6150	0	0	0	0	0
Employee Travel - Out of State	6160	298	1,752	746	5,704	
Student Travel - Out of State	6165				5,881	
Personnel Recruiting - In of State	6350			329	55	
Personnel Recruiting - Out of State	6360		55	1,229	2,459	
Software Maintenance	6196					
Supplies	6201	681	1,116	608	1,432	558
Supplies - Course Fee Funded	6203		0	0	0	0
Software	6211		74			270
Software - Academic	6215			299	60	
Printing off-Campus	6230	86			160	
Subscriptions & Books	6310	208	146	-75	-140	248
Telephone Calls	6401	116	165	142	75	31
Telephone Equipment	6410	2,772	2,772	2,772	2,442	2,376
Postage	6430	0	1	36	690	9
Freight In	6450				400	
Dues & Membership	6470					
Advertising - Searches	6481			780		
Advertising	6480					
Equipment Repair & Maint.	6501					
Maintenance Supplies	6550					
Rent/Lease Equipment	6760	211		453	1,324	1,361
Official Functions Student Related	6830					105
Official Functions	6850	47			76	177
Other Costs	6980					
Professional Development	7150					
Capital Equipment	8090					

Equipment	8100	1,332			58	
Computer Equipment	8101			711	3,084	
Budgeted Change - Rollover	8885					
Transfer PC replacement	8950					
Restricted Reserve - Course specific fees	8980					

Colorado Mesa University
Computer Science Department
External Review Report
Dr. Bob Nielson

Introduction

Colorado Mesa University (CMU) is a regional university with about 11,000 students. According to the Grand Junction area Chamber of Commerce CMU it has a 417 million dollar impact on Grand Junction.

The top areas of business activity at Grand Junction are tourism, healthcare, and education.

CMU has a computer science department with about 25 graduates per year. It focuses on C++ programming skills. Its curriculum is closely matched with the Association of Computing Machinery (ACM) guidelines. The ACM is the de facto standard for computer science programs.

The program's self-study is a realistic and accurate appraisal of the program.
CMU computer science staff and administration has done an excellent job with their self study. It has pointed out several weaknesses and has corrected them. In the previous external review all of the top 5 items were corrected:

1. Retention - Retention was improved by 10%
2. Online courses - Online courses were created.
3. Faculty research - Faculty members participated in the DARPA challenge and several other research activities.
4. Combining computer architecture and assembly into one class. - This was completed.
5. Advanced programming restructure - Done.

Faculty should be praised for following up and doing this.

The program's mission and its contributions are consistent with the institution's role and mission and its strategic goals.
Here is CMU's mission statement:

> Committed to a personal approach, Colorado Mesa University is a dynamic learning environment that offers abundant opportunities for students and the larger community to grow intellectually, professionally, and personally. By celebrating exceptional teaching, academic excellence, scholarly and creative activities, and by encouraging diversity, critical thinking, and social responsibility, CMU advances the common good of Colorado and beyond.

The computer science department is supporting this mission statement by providing excellent classes that support a dynamic learning environment. The department's DARPA challenge, and FIRST robotics involvement help to support scholarly and creative opportunities for both faculty and students.

The program's goals are being met.

Here are the programs goals:

Goal 1: Provide a strong program in computer science.

Goal 2: Prepare students for employment in such fields as software engineering and design, system programming, applications programming, and cybersecurity.

Goal 3: Prepare students to enter graduate programs in computer science.

Goal 4: Provide an opportunity for persons in the geographic region to increase their skills and employability.

These goals are being meet with excellent classes in computer science. The classes being taught are rigorous and challenging. Class are preparing students for employment. Since the classes are ACM aligned, students can easily enter a graduate program.

The curriculum is appropriate to the breadth, depth, and level of the discipline.

Classes are taught at the breadth and depth appropriate to the discipline. Students report that on a Likert scale (1-5), the classes average about 3 in difficulty level. That is right where it should be.

The curriculum is current, follows best practices, and/or adheres to the professional standards of the discipline

CMU curriculum is aligned with the ACM standards. The only class that is part of the ACM curriculum that is not currently being taught is "Theory of Computation". Perhaps this should be added or combined with another class. This class would aid students that move on to graduate programs. Most graduate programs would require students to complete this class before full admission.

Student demand/enrollment is at an expected level in the context of the institution and program role and mission.

Student enrollment in the CS department seems a little low for an institution this size. Other areas of emphasis could increase enrollment. I would recommend a web development and an information technology emphasis be evaluated. Faculty mentioned that there are faculty openings. These emphases could be added with those new hires.

The program's teaching-learning environment fosters success of the program's students.

Teaching seems to be a priority at CMU. There exist several labs and equipment for students to get real experience with embedded systems and software/hardware integration that is not normally seen at an institution this size. Classroom are well equipped with computers and overhead projectors. IT resources exist to meet current demands and future expansions of the department.

Program faculty members are appropriately credentialed.
Here are the key faculty members from the CS department and their academic degrees:

Lori Payne - Phd - University of Northern Colorado
Gary Rader - Phd - University of Pennsylvania
Warren MacEvoy - Phd - University of Arizona
Ram Basnet - Phd - New Mexico Institute of Mining and Technology
Karl Castleton - MS - Washington State University

Key faculty members all have appropriate academic credentials. Most also have extensive "real world" experience. Other faculty members are available (part time or adjunct) with "real world" experience.

Program faculty members actively contribute to scholarship, service and advising.
Most faculty members are involved with community service in some form or another. Faculty members are heavily involved in DARPA, FIRST robotics, and the ACM club.

Campus facilities meet the program's needs.
All classrooms are well equipped with computers for students and faculty to use. All classrooms are equipped with overhead projectors. Campus IT services is well equipped to handle the current and future needs of the CS department. As the department expands additional classroom space will be needed.

Equipment meets the program's needs.
There is also computer hardware for the students to experiment with. Embedded systems hardware and labs are also available for students to experiment with. This is unusual for an institution this size.

Instructional technology meets the program's needs.
Current instructional technology needs are being meet.

Current library resources meet the program's needs.
In a CS department, few library needs exist. Those needs are being met. The library has designated a staff member to acquire any materials needed. However most resources are online. Online resources are being met within the CS department. As the department expands into research, the ACM digital library needs to be acquired.

Student learning outcomes are appropriate to the discipline, clearly stated, measurable, and assessed.

Here are the student learning outcomes (SLO) for the CS department:

A1 – Students will have a working knowledge of a general purpose programming language.

A2 – Given a technical specification, students can develop a software solution to a problem.

A3 – Students understand the powers and limitations of basic computer hardware and software.

A4 - Students can communicate technical concepts.

A5 - Students understand the dynamic nature of computer science.

B1 – Students will have a working knowledge of several programming languages, and the ability to translate concepts between languages.

B2 – Given a problem, students can research and develop the technical specification, and develop, design and test a software solution.

B3 – Students can analyze and measure competing hardware and software components and defend a choice for a given situation.

B4 – Students can compare and contrast competing technical methodologies, explaining and defending choices.

B5 – Students will independently learn and use new technologies.

B6 – Students can work in teams to solve large scale problems.

Outcomes starting with an A are associate. Outcomes starting with a B are baccalaureate level. All outcomes are well defined and mapped to each class.

Program faculty members are involved in ongoing assessment efforts.

Yes. The assessment officer complemented the department's efforts.

Program faculty members analyze student learning outcome data and program effectiveness to foster continuous improvement.

It appears that faculty members are analyzing the student learning objectives (SLO). They are collecting data on those SLO’s. It appears they are using the results to improve the classes and faculty members.

The program’s articulation of its strengths and challenges is accurate/ appropriate and integral to its future planning.

Here are the programs perceived strengths and weaknesses:

Strengths:

Computer Security (including a new certification track)

Mobile Applications (which didn’t exist in 2009)

Embedded Systems (robotics projects and courses and K-12 outreach)

Robotics projects

Maverick Innovation Center

Weaknesses:
Difficult to find faculty
Classroom and Lab resources

The department is making excellent use of current faculty and lab resources but they will be challenged to grow much with their current faculty and lab resources. They are to be complemented with the innovative curriculum and research that they are doing with their current resources.

Recommendation 1 - Class assignments should have a concrete list of specifications to be meet.

In a meeting with students, it was brought up that some assignments are not defined well. Out in the 'real world' there is usually a set of clearly defined specifications that software is to meet. Since CMU is training students for the "real world" this should be emulated.

Recommendation 2 - A meeting with employers in the area should be arranged to make sure that the focus of the classes being taught are what the area needs.

A meeting with employers in the area should be conducted. Current needs of employers should be discussed. Then classes should focus, at least in part, to those needs. This should help aid students get jobs in the area. Some students stated that they would like to find a job in the Grand Valley but find it hard. There are a lot of medical businesses in the area. There are also a lot of tourism businesses. Are their needs being fulfilled?

Recommendation 3 - Other emphases should be explored.

There are a lot of jobs in the IT (Systems Administrator) section of health care. Perhaps an emphases in IT would be appropriate to add. There are also a lot of jobs in web development in the tourism sector. Perhaps an emphasis in web development would also be appropriate to add. These two areas are major job sectors in the Grand Valley business ecosystem.

Recommendation 4 - Consider teaching "Theory of Computation".

This is the only class within the ACM curriculum that is not currently being taught.

Recommendation 5 - Add unit testing to the Software Engineering class.

One of the previous graduates mentioned that unit testing was not being taught. Unit testing is a very popular test methodology being used in current software development. Also consider dependency injection instruction. It is a new and up and coming testing methodology related to unit testing.

Strength 1 - The faculty

The students stated that they love the faculty. Faculty members are friendly and are always available to help the students in whatever problem they face. Faculty members also conduct research in several areas including DARPA. Other faculty members also donate time in the FIRST robotics leagues. They help to recruit new students.

Strength 2 - Research Opportunities

The faculty is actively involved with research. They are involved with the DARPA challenge. The innovation center is available to provide space to conduct research. There are several labs available to students to "play" with hardware and interface C++ programming to hardware.

Strength 3 - Community Engagement

There are also faculty members actively involved in the FIRST robotics leagues. These leagues allow elementary age to high school age students to build a robot and compete to see whose robot is the best. Other faculty members are involved in the ACM organization. These professors are involved with the ACM programming contest. This contest is a fun way to improve the coding abilities of university students.

Strength 4 - Curriculum

CMU has a focus in embedded systems programming that is unique to the institution. Also they have created and are building an emphasis in computer security. Computer security is experiencing major growth in the job market.

CSMS Response to the external review report.

Response to the Summary

In the summary, Dr. Neilson did not agree with the statement, "*Student demand/enrollment is at an expected level in the context of the institution and program's role and mission*." He commented that *enrollment seems a little stagnant.*

To address this point, here is the credit hour production data, updated to include the most recent years.

Academic Year	Student Credit Hours (SCH)	% Change Year over Year	Notes
F11-S12	2716		Anne Spalding on sabbatical.
F12-S13	2798	3.02%	Arun Ektare started transitional retirement (1/2 time).
F13-S14	2694	-3.72%	Hired Ram Basnet tenure track (replacing Arun Ektare), still down 1 position.
F14-S15	2641	-1.97%	Karl Castleton hired full time
F15-S16	2883	9.16%	
F16-S17	3068	6.42%	John Dobbs was hired full-time

In fact, Computer Science has increased significantly: a 9% increase in F15-S16, a 6% increase in F16-S17. We are on track to increase again in the coming year.

The prior decreases have explanations: Computer Science lost two faculty (one through retirement and the other did not return from sabbatical) out of 5 ½ faculty. Classes that could have been taught were not as there was no faculty to teach them. It took three years to replace them.

At the same time, CSCI 106 Web Page Design I, which was our only general education course, experienced significant losses when Applied Studies was dropped from the institution requirements. This caused a decrease from 9 sections and 804 SCH in F10-S11 to 5 sections with 282 SCH in F16-S17. The lack of technology in the essential learning categories seems hard to defend, but the current situation is hurting CS credit hour production. The principle reason the loss was less impactful on overall department numbers than it could have been was that other areas of Computer Science were increasing (specifically in the major). With the acceptance of CSCI 100 Computers In Our Society as a Social/Behavioral Science Essential Learning course for Fall 2015 (first offered Spring 2016), those numbers should increase once again.

Additionally, the comparison between the size of CMU's Computer Science department and Dr. Neilson's department may not be a fair comparison. Dr. Neilson's program is more of a combined program of three programs at CMU: Computer Science, Computer Information Systems, and Information and Computer Technology at the WCCC campus. Comparing CU Boulder with CMU using data gathered from the Colorado Department of Education, in F15-S16 CU Boulder granted 104 CS baccalaureates in a class of 1,859 STEM baccalaureate degrees (5.6%), while we granted 20 out of 163 STEM baccalaureate degrees (12.3%). Comparing across all students, the ratios are 1.4% (CU, 7486) vs 1.3% (CMU, 1472), which is still comparable but does not seem to fairly account for differences in student population. Academic year F14-S15 is the latest year for which we could find STEM data.

http://highered.colorado.gov/Publications/Reports/Degree/fy2015/STEM_sum4yr_fy1315.pdf

http://highered.colorado.gov/Data/Search.aspx

Response to recommendations

Recommendation 1 - Class assignments should have a concrete list of specifications to be meet.

In a meeting with students, it was brought up that some assignments are not defined well. Out in the 'real world' there is usually a set of clearly defined specifications that software is to meet. Since CMU is training students for the 'real world', this should be emulated.

> Computer science, as applied to the real world, is not a field of well-defined problems. In lower division courses, problems are given in well-defined format, fitting for their level. However, as students advance through the program, more open problems are given, allowing the student to be creative and to determine their own constraints. This culminates in the capstone project in CSCI 490 Software Engineering, where they must find and solve a problem of their own choice. Computer Science is one of the most impactful expressions of creativity in the world today, and students must be allowed that experience.

Recommendation 2 - A meeting with employers in the area should be arranged to make sure that the focus of the classes being taught are what the area needs.

A meeting with employers in the area should be conducted. Current needs of employers should be discussed. Then classes should focus, at least in part, to those needs. This should help aid students get jobs in the area. Some students stated that they would like to find a job in the Grand Valley but find it difficult to do so. There are a lot of medical businesses in the area. There are also a lot of tourism businesses too. Are their needs being fulfilled?

Computer Science faculty have significant contact with local businesses, including collaborations, and work with employers to place our students. Many of these companies are owned and staffed by our former students (including the major medical service support company and Rocky Mountain Health Plans). Additionally, we meet with businesses through the Maverick Innovation Center, Grand Junction Business Incubator, and the Grand Junction Maker Space (where one of our faculty is a board member).

We do try to provide resources for them, but quite often our students are paid far better using those skills outside the local area. Our first loyalty is to our students, and much prefer that they work at Amazon, Visa, or Raytheon Systems at 100K jobs than stay in the valley for a 40K job. Some of our students do wish to remain in the valley and are willing to take the pay cut to stay.

Also, many local employers often want fairly low skill workers, like help desk support. This kind of work does not typically require a 4-year degree, and our resource limitations and conflict with the mission of WCCC information technology tracks limit our role with these kinds of jobs.

A small but potentially useful change will be to add a certification in web development using courses we already offer. This may address a web developer track for the contract work associated with tourism and other local industries.

Recommendation 3 - Other emphases should be explored.
There are a lot of jobs in the IT (Systems Administrator) section of health care. Perhaps an emphasis in IT would be appropriate to add. There are also a lot of jobs in web development in the tourism sector. Perhaps an emphasis in web development would also be appropriate to add.
These two areas are major job sectors in the Grand Valley business ecosystem.

Many of our current students are hired to work in these areas, with our department representing a significant fraction of computer technology support in the Grand Valley (anecdotally), perhaps especially in the medical area, which both employs many of our graduates and is serviced by companies started by our graduates. We already have a web development track (although they could be grouped into a certificate), but have real reservations as to how many full time jobs there are in web development in the travel and leisure sector of the Grand Valley.

> Curriculum changes will be proposed in the fall responding to a CS alternative analogous to the Mechanical Engineering Technology degree now offered in Mechanical Engineering. This degree would have lower math requirements, and provide a path to software development (including web and mobile applications), and information technology tracks (including computer security).

Recommendation 4 - Consider teaching "Theory of Computation".
This is the only class within the ACM curriculum that is not currently being taught.

> CSCI 482 Theory of Computation is a course which has not been offered in a long time; the last time it was taught as a regular course was 1997 with 5 students (and later as a topics class with 2 students). It is a theoretically interesting class, and a few graduate schools do require it, but those universities generally allow students to take it as a graduate class. It is difficult to offer at CMU because of staffing limitations and lack of student demand.

Recommendation 5 - Add unit testing to the Software Engineering class.
One of the previous graduates mentioned that unit testing was not being taught. Unit testing is a very popular test methodology being used in current software development. Also consider dependency injection instruction. It is a new and up and coming testing methodology related to unit testing.

> In response to feedback and assessment, in Fall 2016 we began incorporating unit testing into the capstone (CSCI 490 Software Engineering) and testing in general into our required courses such as Advanced Programming in Python, Java, etc.