ABET
Self-Study Report

for the

B.S. Computer Science Program

at

University of Alaska Fairbanks

Fairbanks, Alaska

June 2011

CONFIDENTIAL

UAF College of Engineering & Mines

TABLE OF CONTENTS

BACKGROUND INFORMATION

A. Contact Information

Kara Nance, Chair and Professor of Computer Science
University of Alaska Fairbanks
Department of Computer Science
PO Box 756670
Fairbanks, AK 99775-6670
Phone: 907-474-6104
Fax: 907-474-5030
Email: klnance@alaska.edu

B. Program History

The Computer Science degree at the University of Alaska Fairbanks (UAF) has been in existence since 1985. The program has been accredited by ABET (originally by CSAB) since 1991. All accreditation visits have resulted in the program being accredited for six years. Accreditation visits have been six years apart, (with the exception of an interim visit when CSAB/ABET merged to allow the CS Department to synchronize its accreditation schedule with the other ABET-accredited programs at UAF, which were already synchronized.)

During the last general ABET review, in 2006, the Computer Science Department was a department within the College of Natural Science and Mathematics (CNSM). On July 1, 2010, the Department of Computer Science was transferred to the College of Engineering and Mines (CEM). The move was originally requested by a unanimous vote of the Computer Science faculty. Motivating factors in the formal request memo from the Computer Science Department included the following statements:

ABET Accreditation: CS is the only ABET accredited program that is not in CEM. The overhead and work associated with maintaining ABET accreditation is significant and requires administration that can make informed decisions about the ABET process throughout the accreditation life cycle, including faculty workloads, course offerings, curriculum and program evolution, reporting requirements, and support requirements. Currently CS prepares its own report through CNSM, independent of the CEM ABET accreditation, frequently in a reactionary matter, with no oversight. CEM has staff, administration, and faculty knowledgeable about ABET and a very mature process for ensuring that accreditation requirements are documented and met each year. It is appropriate that CS work with CEM so that duplication of effort is minimized and one cohesive front is presented during ABET accreditation visits.

Computer Engineering: Computer Engineering is a new degree program through CEM that is tightly coupled with the Computer Science program. The program requires that students complete CS 201, 202, 301, 307, 311, 321, and 331 as well as 9 credits of approved electives, which can all be Computer Science courses. While the close ties between the two programs and many associated benefits have been identified, the following deserve special mention:

- Curriculum Development: Both Computer Engineering and Computer Science require joint curriculum development and course offering if both programs are to operate successfully under ABET accreditation requirements to the benefit of the UAF students interested in these majors.
- Lab Facilities: Both Computer Engineering and Computer Science have teaching and research lab facilities that could be shared to the benefit of both departments and all associated students and faculty.
- Robotics: Both Computer Engineering and Computer Science are building capacity in robotics. If the two were in the same college, a unified approach towards developing courses in this high-demand area could be addressed to the benefit of students and faculty.
- Student Cohort Groups: The students in the two majors form cohort groups as they move through classes together. It would be to the benefit of the students to be able to plan course offerings and curriculum development cohesively so that both programs flourish.

Peer Evaluation: Computer Science differs significantly from the departments and programs in CNSM. These differences are most obvious during the peer review process as Biologists and Atmospheric Scientists struggle to understand our research expectations, accreditation issues, faculty teaching loads, and service expectations. Due to the small number of senior faculty in the CS department, it is not unusual for us to be a minority at our own peer review meetings. CEM departments all face the same issue as their department sizes range from 7-11. They have a core peer review committee for all programs that is supplemented by the tenured faculty in each department. We feel that this structure and our close coupling with Computer Engineering will allow us to improve the peer review process for CEM and CS.

C. Options

The following undergraduate degrees are available from the CS Department:

Program Title	Years Required	Degree Awarded	Administrative Unit	If accredited, by whom
Computer Science	4	B.S	Department of Computer Science	ABET
Computer Science	5	B.S./M.S.	Department of Computer Science	ABET (BS Portion only)

Information Assurance Certifications – Based on student demand, UAF Computer Science faculty created a focus in computer security and information assurance. To support this effort, several proposals were written and funded, mainly in support of the creation of the ASSERT Lab and expansion of the curriculum to meet the National Security Agency (NSA) standards in this area. The increase in information assurance offerings been received positively by students and the Industry Advisory Council. The National Security Agency (NSA) has reviewed and approved the UAF information assurance curriculum to grant NSA-approved certificates in Information Assurance at the undergraduate and graduate levels. The certification is coupled with existing degree programs and can be part of the undergraduate or graduate degree. The Information Assurance courses can also fulfill elective requirements for the degree.

D. Organizational Structure

The University of Alaska Fairbanks, www.uaf.edu, with its main campus in Fairbanks, AK, is one of the three universities that together form the University of Alaska (UA) System. The other two are the University of Alaska Anchorage, UAA, with a main campus in Anchorage, AK and the University of Alaska Southeast, UAS, with a main campus in Juneau, AK. Academics at UAF are spread primarily across "Schools" and "Colleges," with "Schools" being narrower in focus and smaller in size than "Colleges." There are four colleges at UAF, including the College of Engineering and Mines, the College of Liberal Arts, the College of Natural Science and Mathematics, and College of Rural and Community Development. There are also four schools, including the School of Management, the School of Fisheries and Ocean Sciences, the School of Natural Resources and Agricultural Sciences, and the School of Education.

Academic Organization

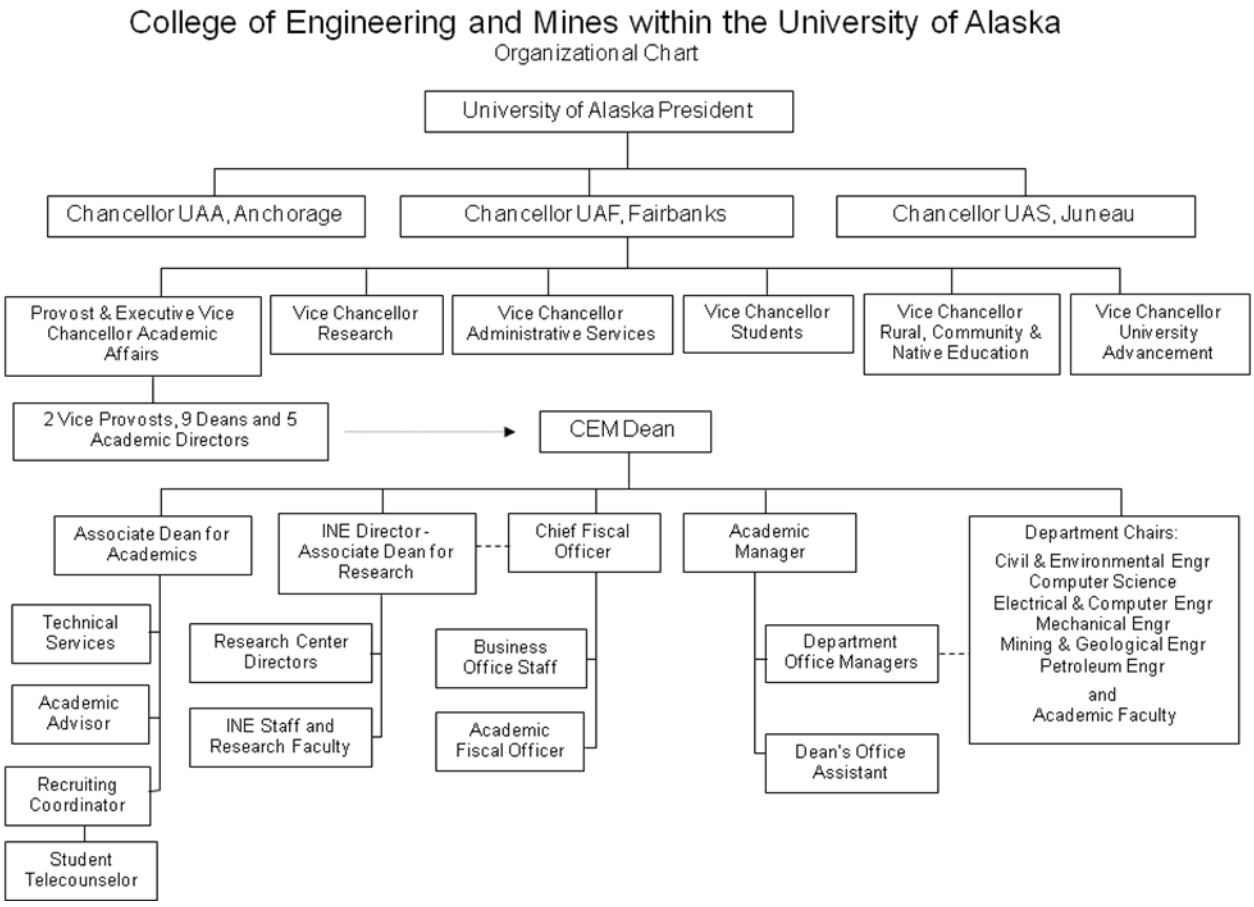
There are six academic departments within the College of Engineering and Mines (CEM), www.alaska.edu/uaf/cem, including the Department of Civil and Environmental Engineering, the Department of Computer Science, the Department of Electrical and Computer Engineering, the Department of Mechanical Engineering, the Department of Mining and Geological Engineering, and the Department of Petroleum Engineering. CEM is led by a Dean and two Associate Deans, one for academics and one for research. The Associate Dean for Research also serves as the Director of the Institute of Northern Engineering.

Research Organization

The Institute of Northern Engineering (INE), ine.uaf.edu, is the research arm of the College of Engineering and Mines. Most college research is conducted through INE, which provides support for proposal preparation and project management for externally funded grants. Within the institute are seven centers in which focused research, development, and testing takes place. INE promotes interdisciplinary and collaborative research and development; promotes partnerships with the natural and social sciences, education, business, geography, natural resource management, and law; promotes outreach; and fosters opportunities for faculty, post-doctoral researchers, and students to engage in research.

The College of Engineering and Mines was formed in 2004 with the merger of the five engineering departments. Prior to 2004, the engineering departments were in two separate colleges, along with other departments. The Computer Science Department joined CEM in 2010.

The following organization chart shows how the college is organized internally and within the UA System.



E. Program Delivery Modes

All required courses for the B.S. degree are offered on the UAF campus in a standard face-to-face format either annually, or each semester. Courses are primarily during the day, but occasionally there are courses offered in the evening.

Support courses: Occasional CS lower division courses are offered at the UAF Community and Technical College site in Fairbanks, Alaska. Distance education and web-based courses are offered through the UAF Center for Distance Education, but these are primarily 100-level CS service courses.

F. Program Locations

All Computer Science courses in the B.S. program are offered on the UAF campus.

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

Concern: The limited data collected raises doubts about the adequacy of effectively assessing the extent to which each program objective is met and whether the assessment

process identifies all needed improvements. A more aggressive student data collection process is warranted. A pattern of regularly collecting data is not in evidence. The sparseness of some aspects precludes sensible modifications to curriculum or pedagogy.

Action: A more aggressive data collection plan was implemented, immediately following the accreditation in 2006. This continues to be a challenge as we have a small number of graduates and they are not consistently responsive to surveys. Our collection methods have been formalized to be consistent with the move the CEM and to leverage their methods so that we can all work together on assessment issues. The new, formalized plan developed during the 2010-2011 academic year has the support of CEM Administration and staff, which will continue to improve the process.

Concern: Although the majority of the faculty members are involved in scholarly activity, there are two faculty members with very limited activities in recent years.

Action: One of the faculty members has since retired. Since the recent move to the College of Engineering and Mines, the workloads and annual activities reports are being evaluated by the new administration. Adjustments in workloads, including increased teaching loads to offset decreased research loads are being assigned. Overall, a more active role is being taken by the CEM Dean to ensure that Computer Science faculty workloads accurately reflect their actual contributions.

Concern: Evidence indicates a number of terminations, resignations, and hires in the last few years. The high turnover rate among this relatively small faculty raises a concern as to whether the faculty support is sufficient to attract and retain high-quality faculty.

Action: The faculty has been stable for the past six years. There has been one retirement and one resulting hire. The new hire was formerly a researcher at UAF and had been teaching in the CS Department since 2002.

H. Joint Accreditation

The program is not jointly accredited or seeking joint accreditation by more than one commission.

CRITERION 1. STUDENTS

A. Student Admissions

There are three pathways for a student to be admitted to the BS Computer Science program. The requirements are:

1. As a first time freshmen, the student needs, at the minimum, all of the following:
    a. High school diploma or equivalent
    b. Pass a high school core of 16 credits with a minimum GPA of 2.5
    c. Must have taken SAT/ACT in the past two years
    d. High school GPA of 3.0 OR
        i. High school GPA of 2.5 AND
        ii. ACT composite score of 18 or SAT total score of 1290 (writing skills included)
    e. Must have completed the following in high school: 4 credits of English, 2 credits of algebra, 1 credit of geometry, ½ credit of trigonometry (an additional ½ credit of advanced math is recommended), 3 credits of social sciences, 1 credit of physics or chemistry, 1 credit of natural sciences, 1 credit of elective. Both physics and chemistry are recommended.
2. As a transfer student, the student needs, at the minimum, all of the following:
    a. If the student is transferring with at least 30 credits, then
        i. Must have left the previous institution in good academic standing
        ii. Must have a minimum GPA of 2.0 in each transferred course
        iii. Transferred course work must be relevant to engineering
    b. If the student is transferring with less than 30 credits, then they must meet freshmen admission standards.
3. As a change of major for a current UAF student in a four-year degree program, the student needs the approval of the department chair. A student in a two-year program cannot change their major into a four-year degree program; they have to apply for admission into the program.

If a student does not meet the requirements, they are placed into a “pre-major” sub-group of the department. This applies even to change of majors. The first two pathways are administered by the Office of Admissions (including making the admission decision), while the last pathway is handled by the Registrar’s Office, with the admission decision resting on the chair of the department.

The number of credits completed determines the class standing for a student.

B. Evaluating Student Performance

UAF requires early grade reports for all freshmen students at the end of 6 weeks. These grades are reported to students on their UA Online account. The goal is to give freshmen early feedback on their performance in all classes. With early grade information students can take appropriate action of seeking help in specific classes, giving more attention to classes where they are not performing optimally, or if necessary withdrawing from a class before the deadline for student-initiated withdrawal. Help is available in several tutoring centers, including the College of Engineering and Mines Tutoring Center, the Math Lab, and others. CEM employs an academic advisor, who concentrates on incoming freshmen and lower division students, but can advise students at all levels. The CEM Academic Advisor can provide guidance to students on their options to appropriately deal with lower than optimum early grade reports.

To remain in good academic standing, UAF requires undergraduate students maintain a cumulative GPA and most recent semester GPA of 2.0 or better.

Students whose cumulative and/or semester GPA falls below 2.0 after each fall and spring semester will be put on academic probation. Students on probation may not enroll in more than 13 credits a semester, unless an exception is granted by the appropriate dean. Probation may include additional conditions, as determined by the dean of the college or school in which the student's major is located. Students on probation will be referred for developmental advising/education and/or to an advising or support counseling center. Removal from probation requires the student's cumulative and semester GPAs to be at least 2.0.

The CEM Academic Advisor communicates with all CEM students on probation after each semester, seeking to guide them on appropriate actions for the student to take, including revising course selection for the following semester.

The UAF registration system implemented “Banner Mandatory Placement” prerequisite and co-requisite verification prior to registration for each course a couple of years ago, with CEM volunteering to be a test college. All registration occurs on line, and students cannot register for courses for which they do not have the proper prerequisites and co-requisites. This process has greatly reduced problems of students enrolling in courses without having the proper prerequisites, a condition which sets them up for trouble and possible failure. There are occasionally extenuating circumstances. CEM has a prerequisite and co-requisite waiver form which is used to document any waived prerequisite or co-requisite, justification for the waiver, including conditions and date conditions, and student, instructor, advisor and department chair signatures.

Banner Mandatory Placement makes the assumption that students will pass their current courses. They are allowed to register for the next semester’s courses that include the current semester’s courses as prerequisites or co-requisites. Currently faculty are asked to print out a list of their class’s students (Class List with Prerequisite Checking) from UAOnline prior to the start of the course. There is an identified loophole in the process and we have asked that UAF rerun the Mandatory Prerequisite and Co-requisite verifications after the current semester ends, but before

the next semester starts, notifying students if they are withdrawn from courses due to not meeting the registration requirements.

Faculty and students also have access to *DegreeWorks*©, an online system that allows them to monitor progress and conduct what-if analysis with graduation. This is in addition to *UAOnline* that allows them to view transcripts online.

Computer science students' progress toward their degree is monitored at a number of levels. Incoming students are placed into the appropriate mathematics and English courses based on SAT, ACT, or placement test scores. Freshman progress reports are collected early in each semester, and students with low early grades are contacted by their resident assistants for extra help. Any undergraduate whose GPA drops below 2.0 is put on academic probation, which requires the student and advisor to prepare an academic improvement plan for submittal to the Dean's office before course enrollment.

Prerequisites are also checked at a number of levels. Each semester, students' planned coursework must be signed off by an advisor before UAOnline allows them to enroll. At the start of the semester, instructors use the UAOnline "Course Listing with Prerequisites" feature to ensure each student has each prerequisite. In exceptional cases when the prerequisite is not needed, such as due to transfer credit course number differences, the CEM Dean's Office has a prerequisite waiver form to document the rationale, which must be signed by the student, advisor, and instructor before enrollment.

C. Transfer Students and Transfer Courses

A transfer student is defined as someone coming into the university with at least 30 transferable semester credits. Transfer students are eligible for admission to a baccalaureate program if they have a 2.0 GPA in their previous course work and left their previous institution(s) in good standing. If applying to a technical or scientific program, students may need to have a higher GPA and proof that they have completed appropriate background courses before they will be admitted. Students transferring into a baccalaureate degree program with fewer than 30 semester hours of transferable credit must also meet the freshman admission requirements. Admission status for students who have attended an unaccredited postsecondary institution will be determined on an individual basis.

Credit accepted at UAF that has been earned from other regionally accredited institutions, through military educational experiences or credit accepted by special approval is considered transfer credit. Where possible, transfer credit is equated with UAF courses. Lists of substitutions within the University of Alaska System are available on page 36 of the UAF Catalog. Standard substitutions from non-University of Alaska institutions are also available in the catalog on page 37. UAF is a member of the Servicemembers Opportunity Colleges (SOC) network. For additional information about the SOC program, contact the Office of Admissions.

© SunGard Higher Education

UAF's transfer credit resource website (uaonline.alaska.edu) is an unofficial reference for undergraduate students who are considering transferring to UAF. An official evaluation of transfer credits may be obtained only after formal application and admission to degree-seeking status with UAF.

In order to serve students who transfer among the three institutions that make up the University of Alaska system, UAF, UAA and UAS have identified fully transferable general education requirements for baccalaureate degrees. Credit for course work successfully completed at one UA institution, which applies to general education requirements, will fulfill the same categories at all other institutions. This applies even if there is no directly matching course work at the institution to which the student transfers. Transfer students from UAA or UAS who have completed all general education requirements in the baccalaureate program prior to transferring to UAF will have completed all requirements for the UAF baccalaureate core. Courses taken to complete the general education requirements at UAA or UAS will meet UAF baccalaureate core requirements according to the current table of substitutions for intra-UA transfers. Completion of the 35-credit lower-division requirements (100- and 200-level courses) of the UAF baccalaureate core meets the general education requirements at the UAA and UAS. More information about transfer credit is available at www.uaf.edu/admissions/undergrad/transfer. The Transfer Credit Resource Database is used to facilitate transfer of other courses.

The UAF Admissions Office evaluates transfer students and course credits, often calling the department chair for specific engineering or computer science course transfer equivalencies. The standard approach for evaluating course equivalency is to compare course syllabi, noting course content, course level, prerequisites, course textbook, and credit hours. Sometimes there may not be a direct one-to-one course transfer equivalency, but there is often a block of transfer courses that can be demonstrated to be equivalent to several UAF courses. This type of block transfer is especially important when students transfer from a university on the quarter system. Each quarter credit hour is equivalent to 2/3 of a semester credit hour.

The student is notified of the condition of their admission in the letter of acceptance, and is asked to provide the missing document(s) prior to beginning their studies at UAF. The student is sent a reminder notice 4-6 weeks prior to the beginning of the term. They are allowed to register for their first term if the document(s) is still missing, but a hold is placed on their registration for the following term. Any student who does not provide the missing document(s) will not be allowed to register, get financial aid disbursements, get transcripts, etc.

Information on transferring to UAF is available to students at online at http://www.uaf.edu/admissions/undergrad/transfer/.

D. Advising and Career Guidance

There are a variety of resources available to students for deciding on a major. The Advising Center's DISCOVER program begins with an aptitude inventory, and uses answers to suggest career areas, majors, schools, and financial aid options. The UAF Major Finder contains descriptions of UAF majors, what courses students can expect to take and what career fields are available with that major. The Career Services office publishes an annual Graduation Report, which provides career and salary information on actual UAF graduates. Advisors are available

to discuss course and career options, and about the availability of majors in students' areas of interest. All resources are available to students at the Advising Center's website (http://www.uaf.edu/advising/).

CEM employs an academic advisor, who concentrates on incoming freshmen and lower division students, but can advise students at all levels. Once engineering and computer science students start taking classes within their department, advising is transferred to the department. Some incoming freshmen go straight to the department for advising and bypass the CEM academic advisor. The CEM Academic Advisor maintains an office with posted office hours and is generally easy to find. He is well trained in many of the questions and situations encountered by incoming freshmen. The duties of the position include the following:

a. Advising students on academic course selection, especially incoming freshmen during the summer months. After students are established in a discipline, they are generally transitioned to department faculty for advising. The advisor position is a 12-month position, so students that visit in the summer or try to register in the summer are generally advised by the CEM Academic Advisor.
b. Helping students with non-academic, as well as academic, issues including housing, financial aid information, university resources for transitioning to college life, study skills workshops, etc.
c. Acting as an early intervention advisor for freshmen who do not perform well in the first few weeks of a semester, as indicated by poor attendance or low homework scores. These students are contacted by the academic advisor to see if something can be done to mitigate the situation.
d. Overseeing the engineering tutoring lab, which includes hiring tutors and maintaining records of use.

Students cannot register without consulting with an advisor. UAF has a central advising operation, the Academic Advising Center. To improve advising across campus, the Academic Advising Center now only sees undeclared majors and general studies students. Students with declared majors are sent to their units for advising. The 12-month availability of the CEM Academic Advisor greatly improves advising for engineering students.

The mission of UAF Career Services (www.uaf.edu/career) is: "The Department of Career Services assists individuals in identifying and implementing career choices. We provide career counseling, job search and internship advising, and on campus employer recruiting to students, alumni, staff and faculty." Career Services is active in providing engineering and computer recruiting events, and holds multiple targeted recruitment/employer events on campus every year.

Within the department, students participate in competitions including paper contests, Collegiate Cyberdefense Competition, and Mathematical Contest in Modeling.

For Computer Science, a list of majors and advisors is printed each year to balance the advising load among the faculty. Frequently, new students are assigned *Computer Science Department* as an initial advisor and it becomes the responsibility of the CS faculty to assign an individual

advisor. In order to facilitate group planning, the Computer Science Advising model was modified in 2007 to enable a tighter tie in between advising and schedule planning. The primary advisor for each major grouping is listed below:

Pre-major Advising – Kara Nance
Honors Program Advising – Chris Hartman
Freshmen and sophomores – Glenn Chappell
Juniors and Senior – Select an advisor based on areas of interest or other personal preferences.

Computer Science students are encouraged to develop a relationship and work with an advisor early to develop a plan through to graduation. This plan is kept on file by the advisor and is utilized in future planning throughout the student's academic career. This helps students with load balancing and assists the faculty in planning course offerings.

By getting to know students individually, advisors can provide letters of recommendation, offer ideas for internships and job possibilities that match students' interests and inform advisees about student opportunities within the department.

The Computer Science department sponsors an Open Advising and Registration Information Session for Computer Science students to provide prospective and returning majors information on scheduling, registration, elective and minor options, graduation plans, and assistance with and submission of forms. This successful model has been adopted by other departments in the College of Engineering and Mines and have proven to be VERY successful in getting to know students, ensuring that they preregister, and maximizing information sharing with respect to future schedules, course offerings, etc. After each event, a report is evaluated by the CS senior faculty to determine if any issues merit further investigation or action. (Examples of such issues have included time conflicts between CS courses and courses in other departments, demand for courses, elective priorities for future semesters.)

Rural Student Services (RSS) provides thorough and comprehensive academic advising to Alaskan Native students. Native and rural Alaska students have the option of requesting an RSS Advisor or an advisor within their major field. RSS works closely with departments to ensure that the students receive the best information available.

E. Work in Lieu of Courses

While it is possible for a student to utilize past professional experience to obtain course credit, this is generally not granted except for general introductory courses when the student has good work experience in the field. Most professional experiences do not cover every aspect of an engineering course, and most engineering courses include calculus-based analysis and design aspects, which is a much deeper level of comprehension than the technician-level experience of most pre-engineering work experience. Other ways to obtain credits for work in lieu of courses include advanced placement (AP) credit, high SAT/ACT scores, and testing out. Advanced placement in certain courses is possible for incoming students provided they have a 3 or above in

the appropriate College Board (CEEB) AP course test from high school. Similarly, students with high SAT/ACT scores in the appropriate category can get credit for ENGL 111X. Students can test out of a few courses through the nationwide College Level Examination Program (CLEP). However, both advance placement and testing out are possible only for a few credits of lower level courses. The UAF Catalog details alternate ways to obtain credit (www.uaf.edu/catalog/current/admissions/transfer_placement.html#Alternate_Ways).

F. Graduation Requirements

UAF's Bachelor of Science in Computer Science degree requires a variety of coursework including the university-wide Bachelor of Science core, a number of required CS core courses including the capstone sequence CS 471-472, as well as approved Computer Science and math electives. Each semester, before registering for classes CS students must prepare a registration form and have it approved by their advisor; this occurs either during the advisor's regular office hours, or at our open advising event held each semester. In addition, UAF has implemented an automated degree requirements checking system, DegreeWorks, which is widely used both by students and advisors. As this is a new system, there are occasionally mismatches, for example when requirements can be fulfilled in alternate ways, or when the catalog's requirements are updated.

During the year that the student plans to graduate, the student fills out an Application For Graduation form, and the UAF Registrar's Office Graduation Department then manually completes a degree audit for that student's catalog year. An analysis of the remaining degree requirements is forwarded to the student and the advisor. In addition a summary sheet for all students planning to graduate that semester is sent to the department each semester.

Prior to graduation, the student transcripts are checked by the UAF Graduation Department in the Office of the Registrar and by the Computer Science Department Chair to ensure that all degree requirements have been met.

The following are degrees awarded:

- Bachelor of Science (B.S.) in Computer Science
- Combined B.S./M.S. in Computer Science

G. Transcripts of Recent Graduates

(Transcripts to be supplied separately.)

CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES

A. Mission Statement

University of Alaska
The University of Alaska inspires learning, and advances and disseminates knowledge through teaching, research, and public service, emphasizing the North and its diverse peoples.

UAF
The University of Alaska Fairbanks, as the nation's northernmost Land, Sea, and Space Grant university and international research center, advances and disseminates knowledge through creative teaching, research, and public service with an emphasis on Alaska, the North and their diverse peoples.

UAF College of Engineering and Mines
The College of Engineering and Mines at the University of Alaska Fairbanks advances and disseminates technical and scientific knowledge through creative teaching, research and public service with an emphasis on Alaska and other high latitude regions. The College promotes students' self motivation to excel and guides them towards professional careers and entrepreneurship in an environment of lifelong learning.

UAF Computer Science
To provide quality education consistent with the requirements of the Computing Accreditation Commission of ABET and the recommendations of the UAF Computer Science Industry Advisory Council, which is responsive to the needs of individual students and the diverse population of Alaska.

B. Program Educational Objectives

CS Program Educational Objectives

The faculty of the Department of Computer Science at UAF provides a positive learning environment that enables students to pursue their goals in an innovative program that is rigorous, challenging and supportive. The B.S. program develops practical skills by emphasizing hands-on experience in the design, implementation, and validation of computer systems in an environment that fosters and encourages innovation and creativity. This approach builds the foundation for the following program educational objectives:

1. Breadth: Graduates will utilize their broad education emphasizing computer science to serve as the foundation for productive careers in the public or private sectors, graduate education, and lifelong learning.

2. Depth: Graduates will apply the fundamental knowledge that is prerequisite for the practice of and/or advanced study in computer science.

3. Professional Skills: Graduates will apply skills for clear communication, effective teamwork, professional attitudes and ethics to succeed as a professional.

The Computer Science Program Objectives can be found online at www.cs.uaf.edu/cs/abet

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

The mission of UAF is to inspire learning, and advances and disseminates knowledge through teaching, research, and public service, emphasizing the North and its diverse peoples. The CS Program Educational Objectives are consistent with the mission of the institution in the following ways:

1. CS Program Educational Objective 1 inspires learning and advances and disseminates knowledge through teaching.
2. CS Program Educational Objective 2 inspires learning and advances and disseminates knowledge through teaching.
3. CS Program Educational Objective 3 disseminates knowledge through public service and inspires learning.

D. Program Constituencies

The program constituencies include the following tightly coupled entities:

Faculty – UAF faculty, which includes alumni, and employers
Students – Current students who will become alumni, employers, faculty, students at graduate schools
Alumni – These include former students who now are faculty, employers, graduate school students and faculty
Graduate Schools – These include the graduate schools that our graduates attend for further education. They also provide faculty and former students have become faculty members at other institutions
Employers – Primary employers of our graduates include IBM, NSA, RDI, UAF (OIT, ARSC, ASF), GVEA, Oil Companies (BP, Alyeska), and Native Corporations

The CS Program Educational Objectives meet the needs of the program constituencies by including them in the review and evolution of the program educational objectives as discussed in Criterion 2E.

E. Process for Revision of the Program Educational Objectives

The CS Program Educational Objectives are reviewed and discussed by the Computer Science faculty at the annual fall meeting. The faculty group includes faculty, alumni, employers, and represents the student perspective. In addition, the CS Program Educational Objectives are reviewed and discussed by the Computer Science Industry Advisory Council

when they are revised. This group includes alumni and employers and provides feedback to the faculty on the appropriateness of the PEOs.

Current CS Program Educational Objectives. The Computer Science Program Educational Objectives were most recently updated during the 2010 – 2011 academic year corresponding to the move from the College of Natural Sciences and Mathematics to the College of Engineering and Mines (CEM) and the ABET expectation that they reflect attributes that are measurable after students have moved beyond the student role and into their professions or pursuit. The evaluation included a review of the existing CEM Program Educational Objectives in order to promote cohesiveness within the new college.

CRITERION 3. STUDENT OUTCOMES

A. Student Outcomes

1. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
2. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
3. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
4. An ability to function effectively on teams to accomplish a common goal.
5. An understanding of professional, ethical, legal, security and social issues and responsibilities.
6. An ability to communicate effectively with a range of audiences.
7. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
8. Recognition of the need for and the ability to engage in continuing professional development.
9. An ability to use current techniques, skills, and tools necessary for computing practice.
10. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
11. An ability to apply design and development principles in the construction of software systems of varying complexity.

The Computer Science Student Outcomes can be found online at www.cs.uaf.edu/cs/abet

B. Relationship of Student Outcomes to Program Educational Objectives

The relationship between student outcomes and program educational objectives are demonstrated in the table below. The X indicates a primary relationship between the student outcome and the CS program educational objectives. The overall educational process is tightly coupled and these do not indicate the only relationships in the table.

	CS Program Educational Objectives		
	Graduates will utilize their broad education emphasizing computer science to serve as the foundation for productive careers in the public or private sectors, graduate education, and lifelong learning.	Graduates will apply the fundamental knowledge that is prerequisite for the practice of and/or advanced study in computer science.	Graduates will apply skills for clear communication, effective teamwork, professional attitudes and ethics to succeed as a professional.
a) An ability to apply knowledge of computing and mathematics appropriate to the discipline.	X		
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.	X	X	
c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired goals.	X	X	
d) An ability to function effectively on teams to accomplish a common goal.			X
e) An understanding of professional, ethical, legal, security and social issues and responsibilities.			X
f) An ability to communicate effectively with a range of audiences.			X
g) An ability to analyze the local and global impact of computing on individuals, organizations, and society.	X		
h) Recognition of the need for and the ability to engage in continuing professional development.	X		
i) An ability to use current techniques, skills, and tools necessary for computing practice.		X	
j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.		X	
k) An ability to apply design and development principles in the construction of software systems of varying complexity.		X	

C. Process for the Establishment and Revision of the Student Outcomes

The Computer Science faculty meet yearly to consider the student outcomes, along with the question of whether additional outcomes are needed, beyond those required. During the 2010-2011 academic year, the faculty approved the ABET-required outcomes, and decided that no additional outcomes were necessary.

D. Enabled Student Characteristics

Student coursework is tied to specific characteristics through our student outcomes assessment process. All the characteristics a) through k) are enabled in the required course CS472 (Senior Project and Professional Practice), which is offered every spring semester. The description of this course, as provided in the current UAF Catalog, is given below.

Students work on group projects in a real computer industry environment and produce appropriate documentation and reports. Nature, ethics and legal considerations of the science profession are discussed with an emphasis on ethics. Additional topics include project management, design methodologies, technical presentations, human-machine interface, and programming team interactions.

CRITERION 4. CONTINUOUS IMPROVEMENT

TABLE 4.1 Assessment Data Collection Processes

		2005-2006		2006-2007		2007-2008		2008-2009		2009-2010		2010-2011		2011-2012		2012-2013		2014-2015	
		Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring	Fall	Spring
Performance Criteria	R											X	X	X	X	X	X	X	X
Exit Surveys	C		X		X		X		X		X		X		X		X		X
Catalog Review	R	X		X		X		X		X		X		X		X		X	
Program Review Analysis	R													X				X	
Capstone Review	C	X		X		X		X		X		X		X		X		X	
Field Test	C	X		X		X		X		X		X		X		X		X	
Employer Surveys	D	X																	
IAB Meeting Summary	C																		

Table 4.1 tracks the assessment processes used since the last ABET visit and the planned collection schedule for the next four years.

A. Program Educational Objectives (PEOs)

The following processes are used to gather data to evaluate the PEOs:

1. Employer Surveys: The employer surveys have been deprecated due to low return rates and a lack of candor on the part of those who did respond. Due to our relatively small program, many employers only have 1 or 2 UAF CS graduates and any information they provide would not be anonymous (for the former students). A few employers have also remarked that their company has a policy against providing feedback via these surveys.

2. CSIAC Meetings: Our goal is to have a face-to-face CS Industry Advisory Council (IAC) meeting every two years. This has been difficult as some members are not allowed by their employers to accept reimbursement and CEM has a policy of not paying for outside travel costs. These meetings have been a source of excellent and candid information about our graduates and provide guidance about current and future needs for CS graduates.

B. Student Outcomes (SOs)

Student Outcomes with Performance Criteria

		Strategies	Assess Methods		Assess Source	Time of Collection
a)	**An ability to apply knowledge of computing and mathematics appropriate to the discipline**					
1	Ability to select the proper data structure to solve a problem	CS202	FE		CS311	Each Fall
		CS311	FE		CS411	Each Fall
2	Ability to determine the efficiency class of an iterative algorithm	MATH307, CS311	FE		CS411	Each Fall

3	Ability to determine the efficiency class of a recursive algorithm	MATH307, CS311	FE	CS411	Each Fall

b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

1	Ability to recognize the complexity class of a problem	MATH307, CS311	MT/FE	CS411	Each Fall
2	Ability to use abstraction to solve a given problem with an existing algorithm	CS311	MT/FE	CS411	Each Fall
3	Ability to create a software design document.	ENGL314	HW/PROJ	CS471	Each Fall

c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

1	Ability to design an efficient algorithm to solve a problem (Use same/similar problem for all levels, but should allow improvement as students gain mastery of CS)	CS201	FE	CS202	Each Fall
		CS202	FE	CS311	Each Fall
		CS311	FE	CS411	Each Fall
2	Ability to measure actual performance on a given architecture	CS201	HW	CS301	Each Fall
3	Ability to design a software system based solely on a Software Requirements Document	CS311	PROJ	CS471	Each Fall
4	Ability to implement a software system	CS311	PROJ	CS471	Each Fall

d) An ability to function effectively on teams to accomplish a common goal

1	Ability to create a software requirements document for a real-world client	CS471	SWR Document in Project Notebook	CS472	Each Spring
		CS471	Peer Eval	CS472	Each Spring
2	Ability to design a large software system for a real-world client	CS471	Design Document in Project Notebook	CS472	Each Spring
		CS471	Peer Eval	CS472	Each Spring
3	Ability to implement and deliver a large software system to a real-world client	CS471	Faculty Eval	CS472	Each Spring
		CS471	Client Eval	CS472	Each Spring
4	Ability to create effective program documentation	ENGL 314, CS471	Project Notebook	CS472	Each Spring
5	Ability to attend team meetings and contribute towards the solution of technical problems.	CS471	Project Notebook (minutes, etc)	CS472	Each Spring
		CS471	Peer Eval	CS472	Each Spring
6	Ability to listen and consider all points of view	CS471	Peer Eval	CS472	Each Spring
7	Ability to contribute effectively to a group presentation	COMM131 or COMM141	Presentation (recorded for assessment)	CS472	Each Spring
8	Ability to create software process documents while following a defined process (e.g. Waterfall, Agile, …)	ENGL 314, CS471	Project Notebook	CS472	Each Spring

e) An understanding of professional, ethical, legal, security and social issues and responsibilities

1	Understand and apply the ACM code of ethics (or similar) and principles underlying them	CS471	MT	CS472	Each Spring
2	Understands and honors the property rights of others (IP, Copyright, etc)	CS471	Project Notebook	CS472	Each Spring
3	Demonstrates ethical decision making	Core PHIL req	MT	CS472	Each Spring

f) An ability to communicate effectively with a range of audiences

1	Ability to write a technical "white paper" (e.g. "How does PCI compliance affect the design and development of a web-based store?")	ENGL 314	HW	CS471	Each Fall
2	Ability to give an effective oral presentation	COMM131 or COMM141, CS all	SWE Case Study Presentation (recorded for assessment)	CS471	Each Fall
3	Ability to create effective program documentation	ENGL 314, SW Construction?	Project Notebook	CS471	Each Fall
4	Ability to create effective software process documents	ENGL 314	Project Notebook	CS471	Each Fall

g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

h) Recognition of the need for and the ability to engage in continuing professional development

1	Ability to work independently on complex problems	CS311	HW	CS471	Each Fall

2	Ability to research problems beyond the material covered in class	ENGL314	HW	CS471	Each Fall

i) An ability to use current techniques, skills, and tools necessary for computing practice

1	Ability to write code without bugs (e.g. Exception Safety)	CS201, CS202	FE	CS311	Each Fall
2	Ability to optimize the performance of a program	CS201	HW	CS301	Each Fall
3	Ability to effectively use a version control system to develop software	CS311	PROJ	CS471	Each Fall

j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

1	Ability to select the proper data structure to solve a problem	CS202	FE	CS311	Each Fall
		CS311	FE	CS411	Each Fall
2	Ability to recognize the complexity class of a problem	MATH307, CS311	MT/FE	CS411	Each Fall
3	Ability to design an efficient algorithm to solve a problem (Use same/similar for all levels, but should allow improvement as students gain mastery of CS)	CS201	FE	CS202	Each Fall
		CS202	FE	CS311	Each Fall
		CS311	FE	CS411	Each Fall

k) An ability to apply design and development principles in the construction of software systems of varying complexity

1	Ability to design a software system based solely on a Software Requirements Document	CS311	PROJ	CS471	Each Fall
2	Ability to implement a software system	CS311	PROJ	CS471	Each Fall
3	Ability to effectively use a version control system to develop software	CS311	PROJ	CS471	Each Fall

1. Performance Criteria

Section 4.B lists the Student Outcomes that are measured via performance criteria. Each performance criteria has an assessment method, assessment source and time of collection. Appendix E provides the rubrics and tripwire values associated with each Student Outcome. These Student Outcomes have been put into place as a result of changes to the ABET standards since the last report and will be fully collected and assessed during the 2011-12 academic year. We were able to collect some assessment data while developing these criteria during the 2010-11 academic year as part of ongoing process improvement. We will assess this data before the start of the Fall 2011 semester.

2. Exit Surveys

An exit survey is given to all students in CS 472 each spring. The survey gathers information including student employment plans and student opinion of curriculum. The results are summarized by the CS Assessment Committee and distributed to all CS Faculty and the CSIAC.

3. Catalog Review

Each fall the catalog contents are reviewed by senior faculty prior to the fall faculty meeting to check for posting errors, needed updates, consistency, and other issues for discussion at the fall faculty meeting. This review serves as the starting point for discussion and updates. (See sections 4.C/D)

4. Program Review Analysis

The Board of Regents requires that all programs be reviewed at least every 5 years. These are meant to be assessments of the quality, efficiency and contribution to the mission and goals of UAF. Last year, UAF reviewed all programs and moved to a 7-year schedule to

align with UAF institutional accreditation. The CS program is next scheduled for review during the 2013-14 and 2015-16 academic years. The results of the review are discussed at the fall faculty meeting.

5. Senior Capstone Project Review

Senior capstone projects have been collected continuously since the beginning of the program. The project notebooks are reviewed by senior faculty to a) identify areas where students may need additional focus and b) identify potential areas for new course development.

6. ETS Field Test

Students in CS 472 (Senior Project and Professional Practice) take the ETS Field Test in lieu of a final exam. The results are reviewed by the assessment committee and discussed at the fall assessment meeting. Motivation for students taking the exam has been a challenge so the information is used primarily for guidance rather than as a definitive tool.

C. Continuous Improvement

The following contains a summary of the course changes that have occurred since the last review. Additional material is available in the ABET review room for visitors.

SUMMARY
COURSE CHANGE DISCUSSIONS AND ACTIONS
2006 - 2011

CS 103 – Introduction to Computer Programming

CHANGE SUMMARY: Modified the prerequisite from MATH 107X or MATH 103X or MATH 161X) to Math placement at the 100-level.

ORIGINATIONS:
Catalog Reviews
Faculty Discussion
Student Discussions

MOTIVATION: (2005) Pre-major and lower division interviews at Advising event indicated that new students who were not ready for calculus (which is the prerequisite for CS 201) do not have a CS option of coursework. Discussion with the instructor (Hay) and faculty determined that completion of 100-level math course was not necessary for success in CS 103. Faculty felt strongly that it was important to have a CS programming option to engage pre-majors and other students who did not have the prerequisites for CS 201.

RE-EVALUATION: (2007) More students have enrolled (and are succeeding) in CS 103.

RE-EVALUATION: (2009) Enrollment has increased to the point that we will create an online version of the course. In addition, we will consider increasing the number of sections offered each semester.

RE-EVALUATION: (2010) Offer two sections of CS 103 in the fall and spring semesters at different times to determine if demand is exceeding course capacity.

RE-EVALUATION: (2011) Enrollment for CS 103 is shown in the table below:

	CS 103 Enrollment							
	Fall 2007	Spring 2008	Fall 2008	Spring 2009	Fall 2009	Spring 2010	Fall 2010	Spring 2011
F2F - 1	21	11	21	20	27	31	27	14
F2F - 2							11	14
DE						14	11	10
	21	**11**	**21**	**20**	**27**	**45**	**49**	**38**

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F103 Introduction to Computer Programming
3 Credits

Programming for non-majors and for those computer science students without the background for CS F201. Concepts of object-oriented programming and algorithm design within the syntax of the JAVA programming language. Prerequisites: Math placement at the 100-level. (3+0)

CS 201 - Computer Science I

CHANGE SUMMARY: Split the catalog description so that CS 201 and CS 202 have separate descriptions.

ORIGINATIONS:

Catalog Reviews
Faculty Discussion

MOTIVATION: (2006) Increased dependence on the online version of the catalog rather than the print version facilitated this split after faculty discussion.

RE-EVALUATION: (2009) Continue to monitor enrollment trends in CS 103 and CS 201 with the possibility of increasing to two sections of the course each fall.

RE-EVALUATION: (2010) Continue to monitor enrollment trends in CS 201 with the possibility of increasing to two sections of the course each fall. CS 103 numbers indicate that this may be needed in future.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F201 Computer Programming I
3 Credits

The discipline of computer science including problem solving, algorithm development, structured programming, top-down design, good programming style, object-oriented programming and elementary data structures. Concepts implemented with extensive programming experience in a structured language and with a group programming project. Prerequisites: One year high school level programming or CS F103 and mathematics placement at the F200-level. (3+0)

CS 202 - Computer Science II

CHANGE SUMMARY: Split the catalog description so that CS 201 and CS 202 have separate descriptions.

ORIGINATIONS:

Catalog Reviews
Faculty Discussion

MOTIVATION: (2006) Increased dependence on the online version of the catalog rather than the print version facilitated this split after faculty discussion.

RE-EVALUATION: (2009) Monitor enrollment trends in CS 201 with the possibility of increasing to two sections of CS 202 in spring semesters.

RE-EVALUATION: (2010) Continue to monitor enrollment trends in CS 201 with the possibility of increasing to two sections of CS 202 in the spring semesters.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F202 Computer Programming II
3 Credits

The discipline of computer science including problem solving, algorithm development, structured programming, top-down design, good programming style, object-oriented programming and elementary data structures. Concepts implemented with extensive programming experience in a structured language and with a group programming project. Prerequisites: CS F201. (3+0)

CS 221 - Introduction to LINUX

CHANGE SUMMARY: Course not selected for offering as scheduled.

ORIGINATIONS:

Catalog Reviews
Faculty Discussion (with TVC also)

RE-EVALUATION: (2009) Consider removing this 200-level support course as TVC* offers a similar course that should meet demand.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F221 Computer Programming II
3 Credits
Offered As Demand Warrants

Introduction to the LINUX operating system including system features, scripting, shell instructions, controlling user processes, maintaining and administering a LINUX system. (3+0)

*NOTE – In 2010-2011 academic year, TVC changed its name to CTC

CS 290 – Internship

CHANGE SUMMARY: The course was deleted.

ORIGINATIONS:
Catalog Reviews
Faculty Discussion

MOTIVATION: (2006) The course was originally created for students who wanted an internship experience but were not sufficiently far enough along in the program to be able to demonstrate the application of CS principles in the internship. Course was created at the suggestion of Dr. Gatterdam. No student has ever enrolled in the course so it is not appropriate to maintain it on the books. Should an individual student request a similar experience, it can be handled through the Independent Study option.

CURRENT CATALOG DESCRIPTION OF COURSE:

NONE

CS 307 - Discrete Mathematics

CHANGE SUMMARY: Monitor MATH 307. Request the MATH Department offer the course in fall, rather than spring if only offering it once each academic year, so that students may take STAT 300 and MATH 307 in different semesters and have enough flexibility to schedule in all of the CS courses in a timely manner.

ORIGINATIONS:

Student Discussion
Faculty Discussion

RE-EVALUATION: (2009) MATH has decreased its offering of the cross-listed course (MATH 307) from fall and spring semester to only offering one semester each year. This decreases the flexibility for CS students to complete the course.

RE-EVALUATION: (2011) Some students insisted they would not graduate on schedule with the new change. Further investigation demonstrated that this change would not delay graduation for any student. We will continue monitoring the course and have instructors willing to teach the course, but cannot increase the number of courses we offer at this time.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F307 Discrete Mathematics
3 credits

Logic, counting, sets and functions, recurrence relations graphs and trees. Additional topics chosen from probability theory. Prerequisites: MATH F201X or permission of instructor. Cross-listed with MATH F307. (3+0)

CS 361 - Systems Security and Administration

CHANGE SUMMARY: Was previously CS 302. Since the middle digit of courses is generally indicative of the course subject class, and we are developing a group of computer security electives, the "6" middle digit designator was assigned to the security courses.

ORIGINATIONS:

Faculty Discussion
Industry Advisory Council recommendation

RE-EVALUATION: (2008) All courses in the security track now have the X6X format.

RE-EVALUATION: (2011) The offering information in the catalog is not consistent with our request nor with our course offering plan. A minor change form needs to be filed as course is scheduled to be offered Spring Even-numbered years rather than Fall Odd-numbered Years.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F361 Systems Security and Administration
3 credits
Offered Fall Odd-numbered Years

Advanced systems programming including privileged instructions and system services, authentication technologies, host-based and network-based security issues. Applications to asynchronous I/O, process control and communication, device drivers and file management. Prerequisites: CS F301. (3+0)

CS F381 Computer Graphics

CHANGE SUMMARY: The second half of this course now includes programmable shaders, which are pervasive in modern graphics. Math prerequisite is now Math 202X (calculus III) or Math 314 (linear algebra).

ORIGINATIONS:

Faculty Discussion
Student Requests

MOTIVATION: (2005) Programmable shaders are a standard tool in computer graphics, and were covered quite successfully in this year's CS 481. After faculty discussions, we agreed to add shader coverage to CS 381.

RE-EVALUATION (2006) Taught a shaders-first version of this introductory course, which turned out to be problematic. The central difficulty in this course is breaking down graphics into managable pieces, and beginning the course with shader-based rendering means students have to simultaneously deal with OpenGL and GLUT interfaces, user interface design, coordinate systems, shader syntax, and the vagaries of modern hardware and drivers. After faculty discussions, we decided to start with fixed-function shaders, and then add in programmability after the midterm.

RE-EVALUATION (2007) Previously, the math prerequisite was Math 314 (linear algebra), which covers vectors and matrices well, but includes substantial additional content that is not needed (eigenvalues, condition numbers, etc). Math 202 (calculus III) covers vectors and matrices well, so starting this year we allowed it as the math prerequisite for this course.

RE-EVALUATION: (2009) Added 3D environment mapping to the plain 2D texturing coverage. Programmable shaders, such as reflect(), make this relatively painless.

RE-EVALUATION: (2011) Faculty discussions about switching languages, from C++ and GLSL, to JavaScript and Processing or WebGL. This would reduce student problems with library and header file dependencies, but doubts remain about software stability and portability.

CURRENT CATALOG DESCRIPTION OF COURSE:

> **CS F381 Computer Graphics**
> 3 Credits
> Offered Fall
>
> Creation of computer-generated images on programmable 3-D graphics hardware. Color, lighting, textures, hidden surfaces, 3-D geometric transformations, curve and surface representations, 2-D and 3-D user interfaces, and the visual modeling of physical phenomena. Prerequisites: CS F202; MATH F202X or MATH F314. (3+0)

CS 421 - Distributed Operating Systems

RE-EVALUATION: (2010) Consider including this course in a reorganization to make it more relevant in the modern cloud computing era. Course has not had enough enrollment demand to offer as scheduled.

ORIGINATIONS:

Senior Notebook Reviews
Exit Surveys
Faculty Discussion

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F421 W Distributed Operating Systems
3 Credits
Offered Alternate Fall

Detailed level study of distributed operating system algorithms, functions and associated implementation. Distributed operating system tuning methods and security. Role of distributed operating systems in net-centric computing. Programming, documentation and evaluation of distributed operating system segments as projects. Prerequisites: CS F321; ENGL F111X; ENGL F211X or ENGL F213X; or permission of instructor. (3+0)

CS 425 - Database Systems

RE-EVALUATION: (2009) Consider making it mandatory as most senior projects require the use of databases. After discussion decided that this was not necessary as enough students were taking the course that we usually have at least one member of each capstone project team that has database experience.

RE-EVALUATION: (2011) Consider including this course in a reorganization to make it more relevant or combining it with other courses and making it mandatory. Most senior projects require database knowledge in order to successfully complete the project. Perhaps consider a three-tier course that can address several of the knowledge areas that students wish had been covered (UI-design, database, and web programming)

ORIGINATIONS:

Senior Notebook Reviews
Exit Surveys
Faculty Discussion

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F425 Database Systems
3 Credits
Offered Spring Odd-numbered Years

Data independence, modeling, relationships and organization. Hierarchical, network and relational data models; canonical schema. Data description languages, SQL, query facilities, functional dependencies, normalization, data integrity and reliability. Review of current database software packages. Prerequisites: CS F311; CS F321. (3+0)

CS 431 - Programming Language Implementation

2008 – Not offered due to insufficient enrollment.

2010 – Not enough students enrolled in the course to offer as a stand-alone course, but students were very interested in the offering. After discussion, the faculty determined that we should attempt co-teaching this course with CS 631 as a stacked offering during the Fall 2010 semester. Instructor thought that would prevent course from being cancelled and would allow students who had enrolled an elective option to take the course.

CURRENT CATALOG DESCRIPTION OF COURSE:

> CS F431 W Programming Language Implementation
> 3 Credits
> Offered As Demand Warrants
>
> Design and implementation of major phases of high level language translators including scanning, parsing, translation, code generation and optimization. Students develop a compiler for a language in a group project which emphasizes good software engineering practices in structured design, testing and documentation. Prerequisites: CS F331; ENGL F111X; ENGL F211X or ENGL F213X or permission of instructor. (3+0)

CS 441 - System Architecture

RE-EVALUATION: (2010) There was a proposal to consider offering course in spring rather than fall. Faculty discussions lead to the decision that continuing to offer it in the fall semester was appropriate at this time. Students need to take either CS 441 or EE 443 (which is offered in spring). Continuing to offer CS 441 in the fall semester gives the students an option each semester to fulfill the degree requirement. In addition, it can be a fall remedial class for graduate students who are not prepared for CS 641.

CURRENT CATALOG DESCRIPTION OF COURSE:

> CS F441 System Architecture
> 3 Credits
> Offered Fall
>
> Computer design fundamentals, performance and cost, pipelining, instruction-level parallelism, memory hierarchy design, storage systems, and vector processing. Prerequisites: CS F321; EE F341. (3+0)

CS 451 - Automata and Formal Languages

RE-EVALUATION: (2007) Taught course with 5 students. Enrollment is too low to continue offering on a regular basis with current number of faculty.

RE-EVALUATION: (2009) Course cancelled due to low enrollment. This can cause a problem since CS 411 or CS 451 is required for graduation. Faculty discussed changing requirement to just CS 411, but no consensus was reached.

RE-EVALUATION: (2010) Advised students during Spring 2010 advising event that they would need to take CS 411 in Fall 2010 if they were planning to graduate in Spring 2011.

RE-EVALUATION: (2011) Course not offered due to low enrollment. Faculty again discussed changing CS 451 to an elective course and requiring CS 411 for graduation. Decision was made to include this change in program changes during next major course reorganization.

CURRENT CATALOG DESCRIPTION OF COURSE:

> CS F451 Automata and Formal Languages
> 3 Credits
> Offered Spring Odd-numbered Years
>
> Finite automata, regular languages, phrase structured grammars, context free language, push down automata, deterministic context free languages, recursive and recursively enumerable languages, Turing machines, decision problems, and undecidability. Prerequisites: MATH F307; CS F201. (3+0)

CS 460 - Introduction to Digital Forensics

CHANGE SUMMARY: New Course. This course was previously offered as a special topics elective. It is now a regular course offered in alternate years. This is one of the courses that leads to DHS/NSA certificate.

ORIGINATIONS:

Faculty Discussion
Student Exit Survey
Industry Advisory Council recommendation

RE-EVALUATION: (2009) All courses in the security track now have the X6X format. Continue planning for specialized tracks within CS. This course will fit into the security specialization.

RE-EVALUATION: (2010) Continue planning for specialized tracks within CS. This course will fit into the security specialization. Plans for specific tracks will be proposed during the 2012 spring semester.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F460 Introduction to Digital Forensics
3 Credits
Offered Fall Odd-numbered Years

Takes a hands-on approach to the forensics examination of computer technology. Focuses on the forensic process, methods, and tools utilized to collect and preserve and examine digital evidence. Course topics include: collection, preservation and examination of evidence from computers including file systems, e-mail and malicious code. Prerequisites: CS F321; or permission of instructor. (3+0)

CS 462 - Intrusion Detection Systems

CHANGE SUMMARY: New Course. This course was previously offered as a special topics elective. It is now a regular course offered in alternate years. This is one of the courses that leads to DHS/NSA certificate.

ORIGINATIONS:

Faculty Discussion
Student Exit Survey
Industry Advisory Council recommendation

RE-EVALUATION: (2009) All courses in the security track now have the X6X format. Continue planning for specialized tracks within CS. This course will fit into the security specialization.

RE-EVALUATION: (2010) Continue planning for specialized tracks within CS. This course will fit into the security specialization. Plans for specific tracks will be proposed during the 2012 spring semester.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F462 Intrusion Detection Systems
3 Credits
Offered Fall Even-numbered Years

Focus on IDS theory and practice and its importance; the origin and resolution of common security threats and vulnerabilities; host and network approaches to IDS implementation; and the legal, ethical, and privacy issues associated with IDS use and policies. Prerequisites: CS F361; or permission of instructor. (3+0)

CS 463 - Cryptography and Data Security

CHANGE SUMMARY: New Course. This course was previously offered as a special topics elective. It is now a regular course offered in alternate years. This is one of the courses that leads to DHS/NSA certificate.

ORIGINATIONS:

Faculty Discussion
Student Exit Survey
Industry Advisory Council recommendation

RE-EVALUATION: (2009) All courses in the security track now have the X6X format. Continue planning for specialized tracks within CS. This course will fit into the security specialization.

RE-EVALUATION: (2010) Continue planning for specialized tracks within CS. This course will fit into the security specialization. Plans for specific tracks will be proposed during the 2012 spring semester.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F463 Cryptography and Data Security
3 Credits
Offered Spring Odd-numbered Years

Specialized study of cryptography and its application in securing data systems, with an emphasis on applied cryptography. Topics include history of cryptography, encryption, digital signatures, authentication, electronic commerce, key distribution and management, private and public key cryptography, and protocols. Prerequisites: MATH F307; CS F311; or permission of instructor. (3+0)

CS 471 - Software Engineering

CHANGE SUMMARY: The courses CS 471 and CS 402 (now CS 472) were changed to be more consistent with a year-long senior capstone sequence. This addressed several issues that were causing some challenges in teaching the courses.

ORIGINATIONS:

Faculty Discussion
Student Discussions
Discussion with ABET visiting team
Student Exit Survey
Industry Advisory Council recommendation

RE-EVALUATION: (2007) We should consider viewing CS 471 and CS 402 (Senior Project and Professional Practice) as a year-long senior capstone sequence. Steps to be implemented over the next few years to make this happen would include changing the numbering to be sequential and making CS 471 a prerequisite to CS 402.

RE-EVALUATION: (2008) Some students have completed the prerequisites and are taking CS 471 (and subsequently CS 402) prior to their senior year. Adding a prerequisite of senior standing to CS 472 should help mitigate this problem.

RE-EVALUATION: (2009) The timeline for completing the senior projects (which includes evaluation of projects) prohibits students from completing projects to the level of depth that the faculty would like. We can distribute the RFP during the fall semester with an October deadline and evaluate the projects as part CS 471. This can be easily tied into the current course content as the requirements component of the Software Engineering course (CS 471) is covered early in the semester.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F471 W Software Engineering

3 Credits
Offered Fall

Introduction to basic software engineering principles, techniques, methods and standards as applied to the engineering of complex software systems. Topics from software system development process models, multiple view system modeling and specification using UML, classification of software systems, project management and legal issues. Prerequisites: Senior standing; CS F311; ENGL F111X; ENGL F211X or ENGL F213X; or permission of instructor. Cross-listed with SWE F471. (3+0)

CS 472 (was 402) - Senior Project and Professional Practice

CHANGE SUMMARY: The courses CS 471 and CS 402 (now CS 472) were changed to be more consistent with a year-long senior capstone sequence. This addressed several issues that were causing some challenges in teaching the courses.

ORIGINATIONS:

Faculty Discussion
Student Discussions
Discussion with ABET visiting team
Student Exit Survey
Industry Advisory Council recommendation

RE-EVALUATION: (2007) See CS 471 discussion. Change course number from CS 402 to CS 472. Make CS 471 a prerequisite for CS 472.

RE-EVALUATION: (2008) Some students have completed the prerequisites and are taking CS 471 (and subsequently CS 402) prior to their senior year. Adding a prerequisite of senior standing should help mitigate this problem.

RE-EVALUATION: (2009) The timeline for completing the senior projects (which includes evaluation of projects) prohibits students from completing projects to the level of depth that the faculty would like. We can distribute the RFP during the fall semester with an October deadline and evaluate the projects as part of the CS 471. This can be easily tied into the current course content as the requirements component of the Software Engineering course (CS 471) is covered early in the semester.

RE-EVALUATION: (2010) Virtually all of the senior projects require database experience or have some underlying database structure. In addition many of the real-world projects require e-commerce experience. Re-evaluate the prerequisite structure and perhaps consider creating a course that covers three-tier architectures to help support these projects. (No specific action taken at this point as students are selecting projects and at least some team members have the prerequisite experience to successfully complete that component of the project.)

RE-EVALUATION: (2011) ENGL 314 (Technical Writing) was added as a degree requirement, but not as a requirement for CS 472. The writing quality in CS 472 is not improving and it was noted that several students hadn't taken it yet or were taking it the same time as CS 472. Add this is a prerequisite for CS 472.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F472 W Senior Project and Professional Practice
3 Credits
Offered Spring

Group projects in a real computer industry environment and produce appropriate documentation and reports. Nature, ethics, and legal considerations of the computer science profession are discussed with an emphasis on ethics. Additional topics include project management, design methodologies, technical presentation, human-machine interface and programming team interactions. Prerequisites: Senior standing; CS F471; COMM F131X or COMM F141X; ENGL F111X; ENGL F211X or ENGL F213X or permission of instructor. (3+0)

CS 480 - Topics in Computer Science

CHANGE SUMMARY: No action taken, but consider for future.

ORIGINATIONS:

Faculty Discussion
Catalog Review

RE-EVALUATION: (2009) Discussed that when Cryptography course was approved, we could remove Cryptography from the list of examples. Not critical, but might be appropriate.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F480 Topics in Computer Science
3 Credits
Offered As Demand Warrants

Topics include, but are not limited to; computational linear algebra, cryptography, parallel algorithm development and analysis. (0+3)

CS F481 Topics in Computer Graphics

CHANGE SUMMARY: With the introduction of CS 482, this course will be reduced to spring even-numbered years, and its content focused more on high-quality rendering: raytracing, shadowing, and global illumination.

ORIGINATIONS:
Catalog Reviews
Faculty Discussion

MOTIVATION: (2007) Faculty discussions of the general reorganization of our graphics curriculum; in particular refocusing this course's coverage by moving simulation material to a new course, CS 482.

RE-EVALUATION: (2008) Taught first reorganized version of this course, covering raytraced graphics, antialiasing, and global illumination, which was successful.

RE-EVALUATION: (2009) After successful first run of simulations course, informal discussions converging around this "Computer Graphics Suggested Elective Package": Computer Graphics: Choose any three of:

CS 381—Computer Graphics
: User Interfaces, OpenGL
CS 481—Topics in Computer Graphics
: Raytracing, Shadowing
CS 482—Simulations in Computer Graphics
: Object Materials and Motion
ART 472—Visualization and Animation: a Maya 3D model & animation course; prerequisite is a 2D pixel editing course (Photoshop)

RE-EVALUATION: (2010) Move forward in Spring 2012 with the creation of the graphics once all courses are permanent and appropriate offering updates to catalog are made in fall 2011.

CURRENT CATALOG DESCRIPTION OF COURSE:

CS F481 Topics in Computer Graphics (m)
3 Credits
Offered Spring

Hardware, software and techniques used in computer graphics taken from topics such as volume rendering, particle systems, shading, image processing, computer aided design, video effects, animation and virtual environments. Prerequisites: CS F381. (3+0)

CS F482 Simulations in Computer Graphics

CHANGE SUMMARY: A new graphics elective covering modern simulation technology, replacing the odd-numbered spring offerings of CS 481. Offered on a trial basis in Spring 2009 (as CS 480) and Spring 2011 (as CS 493).

ORIGINATIONS:

Catalog Reviews
Faculty Discussion
Student Opinion Surveys

MOTIVATION: (2007) Faculty discussions of the general reorganization of our graphics curriculum. Primarily, we noticed that the single upper-division elective CS 481 was covering a more and more broad range of material, from raytracing to modeling to cellular automata, and discussed how to rearrange the material in a more repeatable way.

RE-EVALUATION: (2008) Settled on splitting off a new graphics elective. Initial discussions of how this could become a "graphics specialization" elective package.

RE-EVALUATION: (2009) Taught trial course, covering particle systems, 2D grid simulations including cellular automata and the wave equation, multigrid such as incompressible Navier-Stokes, and finite element. Student opinion data suggested paying more attention on 3D models.

RE-EVALUATION: (2010) Ongoing discussions of 381/481/482 material coverage, approaching convergence on current form.

RE-EVALUATION: (2011) Taught trial course again, with more attention to ingesting and using 3D models from real modeling tools. Course content appears stable, and is ready for a permanent number.

NEXT YEAR (2012) CATALOG DESCRIPTION OF COURSE:

CS 482 Simulations in Computer Graphics
3 Credits
Offered Spring Odd-Numbered Years

Software to simulate physical phenomena for use in interactive visualization, such as particle systems, Navier-Stokes fluid dynamics, and finite element solid mechanics. Includes Lagrangian and Eulerian meshes, stability, and discretization order. For interactive graphics use, high performance qualitatively correct simulations are more useful than high-precision solutions. Prerequisites: CS 381 (Computer Graphics) and PHYS 212X (General Physics)

D. Additional Information

In the Fall, 2011 semester, the catalog review will focus on ensuring that catalog description of the course offerings for the 300 and 400-level electives are consistent with a plan that will support the implementation of TRACK options within the degree. The planned areas of specialization will include Computer Graphics, Computer Security, and General Computer Science. The preliminary track proposal for discussion in Fall 2011 is below and planned to be proposed in Spring 2012.

COMPUTER SCIENCE

B.S. Degree Elective packages (concentrations)

Computer Graphics: Choose any three of:

CS 381—Computer Graphics
CS 481—Topics in Computer Graphics
CS 482—Simulations in Computer Graphics
ART 472—Visualization and Animation (typically Maya 3D animation, prereq. is Photoshop)

Computer Security: Choose three of:

CS 361—Systems Security and Administration
CS 460—Introduction to Digital Forensics
CS 462—Intrusion Detection Systems
CS 463—Cryptography and Data Security

General Computer Science: Choose any CS electives at the 300 or 400 level.

CRITERION 5. CURRICULUM

A. Program Curriculum

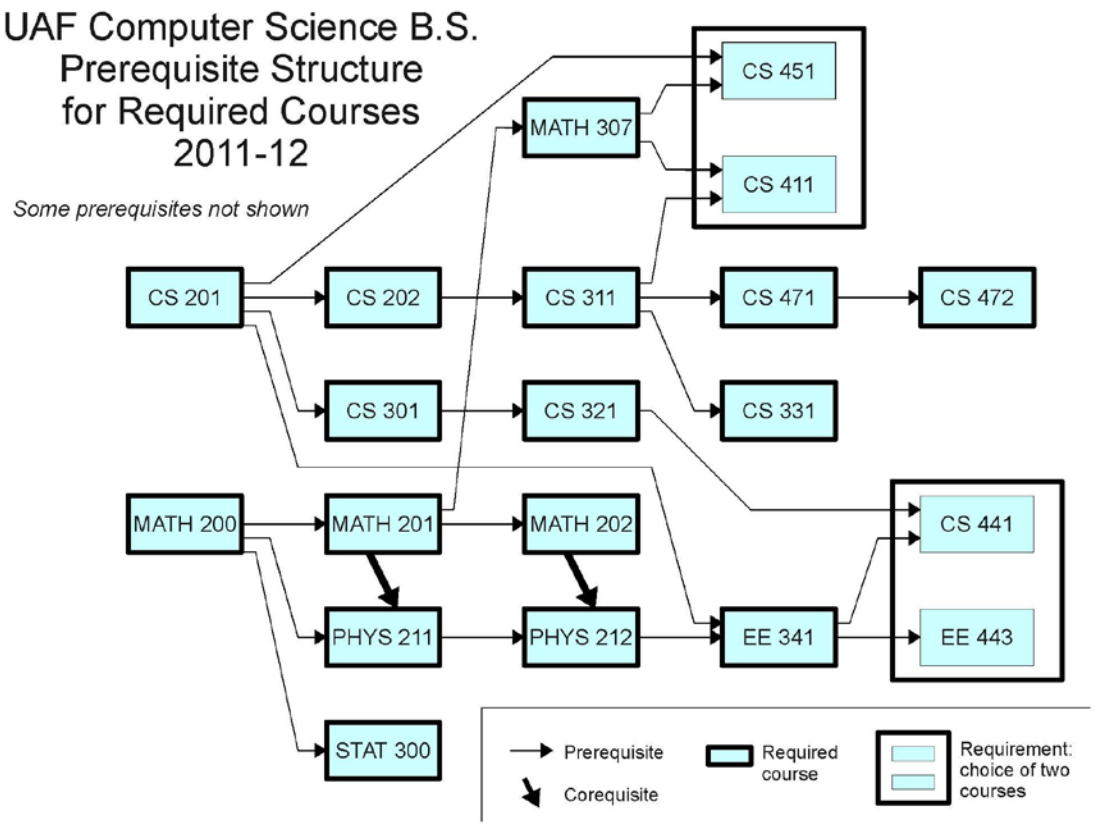
1. Table 5-1 describes the plan of study for students in this program including information on course offerings in the form of a recommended schedule by year and term along with average section enrollments for all courses in the program over the two years immediately preceding the visit. UAF is on a semester system. (See Table 5-1)

2. The curriculum offered aligns with the three major goals of the CS Program Educational Objectives, breadth, depth, and professional skills. Courses are offered in a wide array of subjects, with sufficient detail that students are prepared for the practice of advanced computer science. Professional skills are addressed in such required courses as Technical Writing and the capstone senior project course.

3. The prerequisite structure (shown in 5.A.4) demonstrates the flow of required major courses in the curriculum. The following table shows the a-k criteria and how the courses enable each.

COURSE TITLE	a	b	c	d	e	f	g	h	i	j	k
CS 201	X	X	X						X	X	X
CS 202	X	X	X	X		X			X	X	X
CS 301	X	X	X						X	X	X
CS 311	X		X						X	X	X
CS 321	X	X	X	X		X			X		
CS 331	X			X		X		X	X	X	
CS 411	X		X							X	X
CS 441	X	X	X					X	X	X	
EE 341	X	X	X					X	X	X	
CS 471		X	X			X	X				X
CS 472				X	X		X				

4. Attach a flowchart or worksheet that illustrates the prerequisite structure of the program's required courses.



5. For each curricular area specifically addressed by either the general criteria or the applicable program criteria as shown in Table 5-1, describe how your program meets the specific requirements for this program area in terms of hours and depth of study.

   The general criteria of one year of up-to-date coverage of fundamental and advanced topics in the computing discipline is met by the required CS 471 and CS 472 senior capstone sequence.

   The general criteria of mathematics beyond the pre-calculus level is met by the required math and statistics courses: MATH 200, MATH 201, MATH 307 and STAT 300.

   The program criteria for Computer Science programs is covered in the Program Criteria section.

6. If your program has a capstone or other culminating experience for students specifically addressed by either the general or program criteria, describe how this experience helps students attain the student outcomes.

CS 472 (Senior Project and Professional Practice) has always been the senior capstone course. We previously changed CS 471 (Software Engineering) from an elective to a required course and are in the process of making CS 471 and CS 472 an integrated year-long senior capstone course. The table in Section 4.B lists all of the student outcomes that are measured in CS 471 and CS 472. CS 471 focuses on individual-based student outcomes: some are measures of previously learned skills and some are new skills learned in CS 471 in preparation for CS472. CS 472 focuses on group-based student outcomes and how well the students work together as a team.

7. If your program allows cooperative education to satisfy curricular requirements specifically addressed by either the general or program criteria, describe the academic component of this experience and how it is evaluated by the faculty.

   Methods for earning alternative credit are listed in Criterion 1E.

8. Describe by example how the evaluation team will be able to relate the display materials, i.e. course syllabi, textbooks, sample student work, etc., to each student outcome. (See the 2011-2012 APPM section II.G.6.b.(2) regarding display materials.)

   For each course, there is packet in the team review room. Each packet includes the most recent course textbook, and a notebook containing the following sections: ABET Syllabus (including student outcomes addressed by course), Course Syllabus, Assignments, Graded Assignment Sample (high, medium, low), Exams, Graded Exam Samples (high medium low). In addition, there is a poster in the room that provides a matrix that demonstrates the relationship between courses, rubrics, and outcomes.

B. Course Syllabi

See Appendix B.

Table 5-1 Curriculum

Computer Science

Term	Course (Department, Number, Title) List all courses in the program by term starting with first term of first year and ending with the last term of the final year.	Indicate Whether Course is Required, Elective or a Selective Elective by an R, an E or an SE[2]	*Curricular Area (Credit Hours)*				Last Two Terms the Course was Offered: Year and, Semester, or Quarter	Average Section Enrollment for the Last Two Terms the Course was Offered[1]
			Math & Basic Sciences	Computing Topics Mark with an F or A for Fundamental or Advanced	General Education	Other		
1	Math 200X-Calculus	R	4				*Fall 2010 Spring 2011*	*20*
	CS 201-Computer Science I	R		3F			*Fall 2010 Spring 2011*	*22*
	Engl 111X-Introduction to Academic Writing	R			3		*Fall 2010 Spring 2011*	*24*
	LS 100X-Library & Info Strategies; OR LS 101X-Library Info & Research	R			1		*Fall 2010 Spring 2011*	*50*
	Elective-Natural Science	SE	4					
2	Math 201X-Calculus	R	4				*Fall 2010 Spring 2011*	*33*
	CS 202-Computer Science II	R		3F			*Fall 2010 Spring 2011*	*14*
	Comm 131X-Oral Comm: Group; OR Comm 141X-Oral Comm: Public	R			3		*Fall 2010 Spring 2011*	*23*
	Elective-Natural Science	SE	4					
	Elective	E				3		
3	CS 301-Assembly Language Programming	R		3F			*Fall 2009 Fall 2010*	*15*
	CS 311-Data Structures & Algorithms	R		3F			*Fall 2010 Spring 2011*	*8*
	Engl 211-Academic Writing about Lit. OR Engl 213-Academic Writing about Social & Natural Sciences	R			3		*Fall 2010 Spring 2011*	*22*
	Phys 211-General Physics	R	4				*Fall 2010 Spring 2011*	*75 (Lecture) / 15 (Lab)*
	Core-1[1]	R				3		

4	CS 321-Operating Systems	R		3F			*Spring 2010* *Spring 2011*	*18*
	Phys 212-General Physics	R	4				*Fall 2010* *Spring 2011*	*70 (Lecture) /* *15 (Lab)*
	Math 307-Discrete Mathematics	R	2	1F			*Fall 2010* *Spring 2011*	*25*
	Core-2[1]	R			3			
	Elective	E				3		

5	CS 411-Analysis of Algorithms	R		3F			*Fall 2009* *Fall 2010*	*15*
	Elective-Math[2]	SE	3					
	EE 341-Digital & Comp Analysis and Design	R		4F			*Fall 2009/10*	*11*
	Core-3[1]	R			3			
	Elective	E				3		
6	CS 331-Programming Languages	R		3A			*Spring 2009* *Spring 2010*	*20*
	Stat 300-Statistics	R	3				*Spring 2010/11*	*29*
	Engl 314-Technical Writing	R			3		*Fall 2010* *Spring 2011*	*17*
	Core-4[1]	R			3			
	Elective	E				3		
7	CS 441-System Architecture	SE		3A			*Fall 2009/10*	*10*
	CS 471-Software Engineering	R	3				*Fall 2009/10*	*13*
	Elective-Computer Science[3]	SE		3A				
	Core-Ethics[4]	SE			3			
	Elective	E				3		
8	CS 472-Senior Project / Professional Practice	R		3A			*Spring 2010/11*	*10*
	Elective- Computer Science[3]	SE		3A				
	Elective- Computer Science[3]	SE		3A				
	Core-5[1]	R			3			
	Elective	E				3		
OVERALL TOTAL CREDIT HOURS FOR THE DEGREE	120		35	41	28	21		
PERCENT OF TOTAL								

1. For courses that include multiple elements (lecture, laboratory, recitation, etc.), indicate the average enrollment in each element.
2. Required courses are required of all students in the program, elective courses are optional for students, and selected electives are courses where students must take one or more courses from a specified group.

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

CS Footnotes

[1] Core Courses (15-18 credits) - must complete five listed OR 3 listed plus 2 semester length courses (6 credits) in any non-English language or 3 semester length courses (9 credits) in American Sign Language:
ANTH 100X/SOC 100X
ECON/PS 100X
HIST 100X
ENGL/FL 200X
ART/MUS/THR 200X or HUM 201X or ANS 202X

[2] Math Elective (3 credits) – must complete one: MATH 302, MATH 308, MATH 310, MATH 314, MATH 371, MATH 408, MATH 460.

[3] Computer Science Electives (9 credits) – must be 300/400 level credits or another approved elective (such as EE 443).

[4] Ethics requirement (3 credits) - must complete one: BA 323X, COMM 300X, JUST 300X, NRM 303X, PS 300X, PHIL 322X.

CRITERION 6. FACULTY

Faculty responsibilities, evaluation criteria and workload are governed by 3 documents in order of increasing standing:

1. Faculty Senate policies (www.uaf.edu/uafgov/faculty-senate),
2. UAF Faculty Appointment and Evaluation Policies and UAF Regulations for the Appointment and Evaluation of Faculty (known as "Policies" and "Regulations," and collectively as the "Blue Book"), found at: www.uaf.edu/provost/promotion-tenure, and
3. United Academics Collective Bargaining Agreement (unitedacademics.net).

A. Faculty Qualifications

A full-time faculty workload is 30 units for 9 months, as defined in the United Academics Collective Bargaining Agreement (UNAC CBA, unitedacademics.net). A typical tripartite faculty workload would consist of 60% teaching, 30% research, and 10% service (18, 9, and 3 workload units respectively). Teaching credit consists of formal instructional classes, advising undergraduate students, mentoring graduate students, etc. Research activities include all professional activities leading to publication, performance or formal presentation in the unit member's field, or leading to external funding recognizing the unit member's current or potential contribution to their field. Such activities include manuscript submission, grant proposal submission, supervision of externally funded research projects, development of patentable inventions, additions to a portfolio, and other contributions appropriate to the unit member's field. Service activities include professional service, public service and university service. Typically a faculty member will serve on some committee within the department, college or university. A 3-credit class is worth 3 workload units. A typical workload will consist of four 3-credit courses per year, graduate student supervision, undergraduate student advising, research activities and service activities in all three service categories.

The composition of professional duties and responsibilities of unit members will be determined by the appropriate administrator after consultation with the department head/chair and unit member. Faculty members consult the department chair in writing their proposed workload (www.uaf.edu/provost/faculty-reports-forms/faculty-workload-forms), who then submits workloads to the dean, who modifies the proposals if necessary to achieve the overall balance of required work of the college. It is possible to buy out of a course (3 units of workload for 6 weeks of salary) using research funding or internal competitive grants, provided the academic mission of the college can still be effectively delivered.

One faculty member retired during the year of the last accreditation visit (2005-2006). The new hire is a replacement for that position. All faculty hold 100% appointments in the department. All have Ph.D. degrees in Computer Science or a closely related area. See Table 6.1 and Appendix E for faculty experience.

Table 6.1

Year	Total Faculty	Resignations	Retirements	New Hires
2010-2011	7	0	0	0
2009-2010	7	0	0	0
2008-2009	7	0	0	0
2007-2008	7	0	0	0
2006-2007	7	0	0	1

B. Faculty Workload

Faculty workloads are distributed to allow for sufficient time to attend professional meetings throughout the academic year, and to maintain active research programs. The typical distribution for full-time faculty is teaching 60%, research 30% and service 10%. The normal teaching load is 4 courses per year. The workloads are being adjusted as described in Background Section to address concerns expressed in the previous ABET visit. Tenured faculty are eligible for sabbatical leaves every 6th year. The 2010-2011 academic year had some course releases for faculty members who were involved in an NSF project that required extensive travel.

C. Faculty Size

With the move to CEM, there is additional assistance with student advising, which is helpful, particularly in the summer hours when faculty are not available to assist students. As the graduate programs grow, there are some challenges in course offerings and also faculty time with respect to advising graduate students and meeting the needs of the undergraduate students.

D. Professional Development

The Office of Faculty Development, located at 222 Bunnell, provides professional development opportunities for all faculty members at UAF in the areas of teaching, learning, and scholarship. Assistance with travel, mentoring, promotion & tenure, teaching observations, and instructional technology (through Campus Technology Services) are some of the programs. The office also brings national speakers and trainers to campus, conducts training workshops, and maintains a collection of resource materials on these topics, both in the office and at the Rasmuson Library. Regular workshops, panel discussions, and seminars will be held throughout the year for faculty. Although these are mainly designed for new faculty, they are open to all faculty members, and can be audio-conferenced to the rural campuses if requested. The limited travel funds are generally awarded competitively to new faculty.

Training in other areas such as safety and ethics are also provided through different UAF departments.

CEM provides limited funding on a competitive basis to faculty for professional development activities in academic areas, which include attending short courses that are useful in enhancing the value of courses taught, attending professional conferences, accreditation workshops, and other areas. INE provides limited funding on a competitive basis for faculty professional development in research areas. Additionally, each year CEM has provided departments with about $10K of travel funding to be used for academic development, including ABET training and development.

The College of Engineering and Mines distributes travel awards of up to $2,500 twice per year via a competitive written application process. The CS department's yearly budget has $4,500 - $15,000 allows for travel which is distributed based on competitive requests. All requests for funding were awarded in 2010-2011. Finally, funded research grants typically are budgeted with some amount of travel to disseminate the research results. Together, these provide faculty with the resources they need to attend conferences and workshops, training and networking, and ensure their professional development.

E. Authority and Responsibility of Faculty

The program faculty has control over course content, including creation, modification, and evaluation of courses. However, course creations and modifications have to be approved by the department chair, CEM Curriculum Council, CEM Dean, Faculy Senate Curriculum Council, and the Provost. Every signatory technically has the same responsibility – ensuring that course proposals minimize content overlap, have resources for effective delivery, have a sound teaching and assessment plan, and are compliant with Faculty Senate guidelines. However, some signatories emphasize certain aspects more than the others. The department chair ensures that the course is consistent with the mission of the program and department, and that resources (faculty, lab, etc.) exist to deliver the course effectively. The CEM Curriculum Council ensures that there is no course duplication within the college and looks at the expected rigor of the course, including contact hours versus credit hours, level of material matching the course number, and that all required content is present and clear on the proposed syllabus, etc. The CEM Dean offers guidance in curriculum development to both the department and Curriculum Council regarding realistic constraints of course enrollment numbers, department faculty capacity, number of courses in the degree program, etc., and must agree with the proposal in order for it to go forward. The Curriculum Council of the Faculty Senate examines course proposals in the context of the entire university for duplication, and also ensures compliance with Faculty Senate and university-wide guidelines on course content, syllabus content, and assessment.

College faculty also examine, modify and vote on approval of the CEM Unit Criteria, which are the published guidelines used in the evaluation of CEM faculty during the promotion and tenure process (http://www.uaf.edu/files/provost/CEM-unit-criteria-5-1-06.pdf). These Unit Criteria must be approved by CEM faculty at least every 5 years, and change can be initiated by the CEM faculty at any time, but changes are subject to a approval by vote of the faculty. The same Unit Criteria are used in pre-tenure (mandatory 4th year evaluations) and post-

tenure evaluations of faculty. Unit Criteria are additional to the criteria outlined in the UAF "Blue Book" polices and regulations.

Table 6-1. Faculty Qualifications

Computer Science

Faculty Name	Highest Degree Earned- Field and Year	Rank [1]	Type of Academic Appointment [2] T, TT, NTT	FT or PT [4]	Years of Experience			Professional Registration/ Certification	Level of Activity H, M, or L		
					Govt./Ind. Practice	Teaching	This Institution		Professional Organizations	Professional Development	Consulting/summer work in industry
Chappell, Glenn	Ph.D., 1996	ASC	T	FT	0	22	10	None	L	L	L
Genetti, Jon	Ph.D., 1993	ASC	T	FT	5	24	12	None	M	M	M
Hartman, Chris	Ph.D., 1997	ASC	T	FT	3	13	13	None	L	L	L
Hay, Brian	Ph.D., 2005	AST	TT	FT	10	8	8	None	H	H	H
Knoke, Peter	Ph.D., 1968	ASC	T	FT	30	27	22	PE, CSDP	M	M	L
Lawlor, Orion	Ph.D., 2004	AST	TT	FT	6	5	5	None	L	M	L
Nance, Kara	Ph.D., 1991	P	T	FT	10	25	18	PMP	H	H	M

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

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

Table 6-2. Faculty Workload Summary

Computer Science

Faculty Member (name)	PT or FT[1]	Classes Taught (Course No./Credit Hrs.) Term and Year[2]	Program Activity Distribution[3]			% of Time Devoted to the Program[5]
			Teaching	Research or Scholarship	Other[4]	
Chappell, Glenn	FT	Computer Graphics (CS 381/3cr) Fall 2010	60	30	10	100%
		Complexity of Algorithms (CS 611/3cr) Fall 2010				
		Operating Systems (CS 321/3cr) Spring 2011				
		Programming Languages (CS 331/3cr) Spring 2011				
Genetti, Jon	FT	Analysis of Algorithms (CS 411/3cr) Fall 2010	60	30	10	100%
		Programming Language Implementation (CS 431/3cr) Fall 2010 (Stacked with CS 631)				
		Software Engineering (CS/SWE 471/3cr) Fall 2010				
		Programming Language Implementation (CS 631/3cr) Spring 2011 (Stacked with CS 431)				
		Senior Project and Professional Practice (CS 472/3cr) Spring 2011				
Hartman, Chris	FT	Computer Science II (CS 202/3cr) Fall 2010	60	30	10	100%
		Data Structures and Algorithms (CS 311/3cr) Fall 2010				
		Computer Science I (CS 201/3cr) Spring 2011				
		Data Structures and Algorithms (CS 311/3cr) Spring 2011				
Hay, Brian	FT	Database Systems (CS 625/3cr) Fall 2010	40	50	10	100%
		Intrusion Detection Systems (CS 462/3cr) Spring 2011 (Stacked with CS680)				
		Advanced Intrusion Detection Systems (CS 680/3cr) (Stacked with CS 462)				
Knoke, Peter	FT	Computers and Society (CS 101/3cr) Fall 2010	60	30	10	100%

		Computers and Society (CS 101/3cr) Fall 2010				
		Software Engineering Requirements (CS/SWE 673/3cr) Fall 2010				
		Computers and Society (CS 101/3cr) Spring 2011				
		Advanced Software Engineering (CS/SWE 671/3cr) Spring 2011				
Lawlor, Orion	FT	Assembly Language Programming (CS 301/3cr) Fall 2010	60	30	10	100%
		Computer Architecture (CS 441/3cr) Fall 2010				
		Simulations in Computer Graphics (CS 482/3cr) Spring 2011				
		Advanced Computer Architecture (CS641/3cr) Spring 2011				
Nance, Kara	FT	Software Project Management (CS 602/3cr) Spring 2011	30	50	20	100%

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

CRITERION 7. FACILITIES

A. Offices, Classrooms and Laboratories

Summarize each of the program's facilities in terms of their ability to support the attainment of the program educational objectives and student outcomes and to provide an atmosphere conducive to learning.

1. Offices

All full-time faculty members have their own office space, with adequate desk and shelving space for a computer, peripherals, storage and work area. Offices are equipped with extra chairs for students. All teaching assistants have new furniture (2010). Dedicated Computer Science Conference space is available in the building for small (< 14) or large meetings in the Chapman Building or close by on campus in the rare cases of a scheduling conflict or for larger groups.

2. Classrooms

Classrooms typically include a laptop-associated projector, an overhead projector and a blackboard or whiteboard. Where a laptop-associated projector is provided, faculty normally use a personal laptop.

Two classrooms within the Computer Science Department are "Smart Classrooms", equipped with video/data projector, television set, VCR and DVD player, notebook computer connections (A/V, LAN, and Internet), visualizers, intra-classroom connections, and access to the video distribution hub that allows playback of multiple format videotapes, satellite downlinks, and 16-mm film footage.

The university also supports ten "Smart Carts" for use in classrooms without built-in "smart" equipment. These smart carts include overhead projectors, VCR and DVD players, desktop computers, and notebook computer connections (A/V, LAN, and Internet). In areas where physical network connections are not available, instructors can connect to the Internet using wireless network cards.

3. Laboratory Facilities:

Institutional facilities:

COMPUTER LABS: Campus Technology Services (CTS), under the Office of Information Technology (OIT), mans and maintains computer labs for the UAF campus. These labs are available for use by all UAF students, staff and faculty. Some individual departments have departmental labs, but these are usually restricted to users within those departments. Each lab contains both Windows and Apple computer systems, laser printers, DVD drives, a wide variety of software and related equipment. Consumers who need high-speed Internet access may connect using wired access ports or wireless access points. Although there is no per-visit fee to use the computing labs, consumers are assessed a fee for supplies, such as CD/DVD discs or printer paper. Currently, CTS is responsible for three open labs on campus: the Bunnell Lab, the MBS Lab (24 hour), and the Rasmuson Lab (24 hour). CTS is also responsible for a training lab located in the Rasmuson Library.

The Bunnell Open Lab is the open to all students, faculty, and staff at UAF. In addition to machines running Macintosh and Windows operating systems and a variety of software, there is a flatbed scanner equipped to scan pictures, slides, and negatives as well as text documents available for lab use. Printing is available for both black & white, and color output. There is wireless network connectivity available. Lab consultants are provided to help with questions or problems.

Computer	Processor/RAM	Extras
PC	2.33Ghz / 2GB	DVD-R/RW, USB 2.0, Firewire 400
I-Mac	2.4GHz / 2GB	DVD-R/RW, USB 2.0, Firewire 400/800, Webcam, MS Office 2008, iLife, Adobe CS4

The MBS Open Lab is the open to all students, faculty, and staff at UAF. Black & white printing is available. There is wireless network connectivity available. Lab consultants are provided from 8AM to 5PM to help with questions or problems.

Computer	Processor/RAM	Extras
PC	2.13Ghz / 2GB	DVD-R/RW, 3.5" Floppy Disk, USB 2.0, Microsoft Office 2007, Adobe Suite CS3, Camtasia Studio 5
I-Mac	2.0 Ghz / 2GB	DVD-R/RW, USB 2.0, Firewire 400, Microsoft Office 2008, Adobe Suite CS3, iLife, Blender

The Rasmuson Open Lab in the Rasmuson Library contains both IMacs and Windows based computers and is open to students, faculty, and staff of the UAF campus. Black & white printing is available. Wireless and walk-up network ports are available in the 24-hour study area. Lab consultants are provided from 8AM to 5PM to help with questions or problems.

Computer	Processor/RAM	Extras
PC	2.13 Ghz / 2GB	DVD-R/RW, 3.5" Floppy Disk, USB 2.0, Microsoft Office 2007, Adobe Suite CS3, Camtasia Studio 5
I-Mac	2.0GHz / 2GB	DVD-R/RW, USB 2.0, Firewire 400, Microsoft Office 2008, Adobe Suite CS3, iLife, Blender

The Rasmuson 301 Training Lab is available for all University faculty and staff. The lab is available for scheduling during the library's operating hours. This lab does not have a technician dedicated to supporting this room. For support, there is a phone in the room that can be used to call the OIT Support Center. The following software is available: Microsoft Office 2007, Adobe CS5, ALEKS, Camtasia Studio 5, Citrix, DegreeWorks, OnBase. The lab is equipped with black-and-white printing, wireless network connectivity, a ceiling-mounted LCD projector with XGA native resolution, a stereo speaker system, and a DVD/VCR player.

Computer	Processor/RAM	Extras
PC	3.0Ghz / 2GB	DVD-R/RW, USB 2.0, eSATA Microsoft Office 2007, Adobe CS5, ALEKS, Camtasia Studio 5, Citrix, DegreeWorks, OnBase
Mac Mini	2.0Ghz / 2GB	DVD-R/RW, USB 2.0, Firewire 800

Departmental facilities:

Chapman 103 - Chapman Computer Lab
This lab consists of 20 workstations, two servers, a printer, a scanner, and an overhead display system. One server provides authentication and authorization services for the lab users, including storage for roaming profiles and home directories (at least 50MB per student). The second server acts as lab image and management server (allowing hard disk images to be easily updated and pushed out to the lab workstations, or to manage configuration changes or software update/installations for the lab workstations).

The overhead projector is connected to one of the workstations at the front of the room for in-lab instructional use, and is also connected to a second wall-mounted VGA input to allow instructors to easily connect a laptop to the lab projector. The printer is available on the network from all of the lab workstations. The scanner is connected to a workstation in the back of the room and is provided for students to use in connection with their assignments and projects as they see fit.

Each workstation in the lab currently runs Windows 7, and includes the following software:

- Development Tools
  - MS Visual Studio 2010 supporting development in VB, C++, C#, and F#.
  - Code Blocks 8.02 supporting development in C and C++.
  - Dr Java (latest version) supports development in Java.
  - Cygwin with GCC installed.
  - Fortify Build Monitor
  - Eclipse 3.4.0 for development in Java and C++.
  - Parse Generator for development in Lex and YACC.
  - Python versions 2.7 and 3.1 for development in Python.
  - Perl version 5.10.
  - MS SQL Server 2005 and 2008.
  - NetBeans IDE 6.7.1 for development in Java.
  - Notepad++ for general purpose programming.
- Office Tools
  - MS Office 2007 including Word, Excel, PowerPoint, Access, Project, Visio, Groove, InfoPath, OneNote, Publisher, and Outlook
  - Open Office 3.2 including Base, Calc, Draw, Impress, Math, and Writer
  - Adobe Reader X
- Virtualization Tools
  - vSphere Client, providing a connection to the virtual machines running in the ASSERT RAVE Lab.
- Digital Forensics Tools
  - AccessData's FTK 1.8 for digital forensics.
  - AccessData's Registry Viewer for digital forensics.

Software
The department has licenses for the following academic programs, allowing students, staff, and faculty members within the department free access to their offerings:

- MSDN-AA – This provides access to a wide variety of Microsoft products, including operating systems, software packages (e.g., Exchange server, Visio, MS Project), and development tools (including the latest releases of the Visual Studio suite). All of these products can be used by students for class/research related activities, including the installation of operating systems on their personal workstations/laptops.

- VMware Academic Program (VMAP) – This provides students with access to the vast majority of VMware products (including VMware Workstation, VMware Fusion, vCenter Server, Lab Manage, and ESX), which can be installed on their personal workstations/laptops during their association with the department.

Accounts in the Chapman Computer Lab are available upon request to any student that has paid the semester-based lab fee. Lab fees are collected from students enrolled in any Computer Science course. Students not taking one of these courses may voluntarily pay the semester based lab fee to obtain an account in the lab. Accounts expire, and are archived, after fee payment closes in the first fall or spring semester during which the student has not paid the semester-based lab fee. Archived accounts are kept for up to one year as space permits. Upon request, arrangements can be made to provide students with a copy of their archived data.

Chapman 203
Chapman 203 contains a machine purchased by the UAF Linux User's Group student organization, which is used to stream the university radio stations KSUA and KUAC over the web.

Chapman 205 – Visualization Lab
Room 205 holds the UAF CS Bioinformatics Powerwall. This is a twenty-screen, 8400x4200 resolution, 35.28 megapixel display wall driven by ten NVIDIA GeForce GTX 280 graphics cards and interconnected with gigabit ethernet. The powerwall serves as the testbed a student accessible parallel programming environment for UAF CS developed software packages MPIglut, cudaMPI, glMPI, and EPGPU.

Also in room 205 is the NetRun cluster. This is a diverse collection of hardware, including GPUs and modern quad-core Intel machines, ARM, older PowerPC, MIPS, and SPARC machines, and even a venerable Intel 486. All the machines can be programmed, benchmarked, and compared via a common web-accessible interface "NetRun," which is also used for online grading of homework assignments in several CS courses. This cluster is most useful in computer architecture courses, to illustrate the dramatic differences in performance and assembly code between different generations of hardware. NetRun is used an average of several hundred times per day.

Chapman 206 - ASSERT LAB
The Advanced System Security Education, Research, and Training (ASSERT) Center is located in the Computer Science Department. The ASSERT Center has four lab environments, three of which are physically located in the Chapman building itself

ASSERT SCADA Lab: This environment includes a networked control system, a control workstation, an "attacker" workstation, and a configurable network, and is used to control a simulated petroleum pipeline. The system can be configured to provide the attacker with a variety of vantage points from which to interact with the control system. The lab has been used for research projects in the field of control system security, and to provide students with an understanding of the security implications of network accessible control systems.

ASSERT Digital Forensics Lab: This environment contains a variety of equipment for use in digital forensics, including write blocking devices, cell-phone imaging devices, specialized software, and workstations capable of processing and storing large volumes of digital evidence. There are also several "target" systems that can be used for digital forensics exercises, including hard drives, cell phones, and handheld devices (e.g., iPod touch). This equipment is available to students for research and during classroom exercises. In addition, a separate set of equipment is maintained is a separate area for use during actual civil and criminal investigations in cooperation with law enforcement agencies.

ASSERT Security Lab: This environment features several workstations and network components that can be used to build physical lab environments (primarily for security exercises). It has been used most recently by students in preparation for the Collegiate Cyber Defense Competition (CCDC).

ASSERT Remote Access Virtualized Environment (RAVE): The ASSERT RAVE consists of several servers that support virtualization, and is housed in the UA datacenter. The lab supports the deployment of several hundred virtual machines that that can be assigned to students or researchers. The typical use for this lab is to deploy a set of 2-5 virtual machines to each student in a class, commonly for a security or digital forensics related exercise. Each user can remotely access their own set of virtual machines (although in the case of almost all exercises, the virtual machines themselves are placed on isolated virtual networks to ensure that lab activities cannot impact production networks). In addition to students at UAF, the lab regularly hosts classes from other institutions around the nation, and is currently the model being used as part of an NSF-funded deployment of computer security infrastructure around the nation. While security exercises are the primary focus of this lab, it has also been used by a variety of other classes and groups at UAF and other institutions (including the local community college), including:

- System administration classes (each student was given several virtual machines and networks to manage),
- Database classes (each student was given their own enterprise database server and clients),
- Operating systems classes (each student is given their own virtual machine on which they could explore the operating system code),
- Networking classes (each student was given a network of hosts and routers which they could experiment with)

The ASSERT RAVE lab provides several advantages to a physical lab, including the ability to access it remotely, the quick deployment of machines to students, the ability to provide every student with an identical environment, the ability for group collaboration, and the snapshot capability that allows students to instantly revert their system to a known state. In addition to specific classes that use the lab, it is also used for demonstrations and modules that are integrated into other classes through the computer science curriculum, or

for presentations to other parts of the Alaskan community, including government, business, and the public.

In addition to the specific facilities listed above, the department hosts a departmental web server, and an SVN repository available for faculty, researcher, and student use.

Duckering 208: Electrical Engineering Digital Hardware Lab

The Digital Hardware Laboratory is equipped with 8 identical workbenches. Every workbench provides space for up to three students to conduct basic digital logic experiments. The EE341/EE343 lab exercises conducted in the Digital Hardware Laboratory are based on the C.A.D.E.T. II digital logic trainer, which consists of a breadboard area with surrounding function generator, buttons, switches, LEDs, etc. The C.A.D.E.T. II trainer is used to introduce students to the fundamentals of hardware design before they use advanced, software-based techniques in the Digital Computational Laboratory. Every workbench also has an oscilloscope, multimeter, power supply, and PC. This lab is used for the class taught to CS students, EE 341, as well as the class taught to EE students, EE 343.

Duckering 210: Electrical Engineering Digital Computational Lab

The Digital Computational Laboratory is equipped with 12 PC workstations and 8 new MSO (Mixed Signal Oscilloscopes) which have 2 analog channels and 16 channels of digital inputs for state levels, timing diagrams and view control buses. The laboratory is used to teach state-of-the-art digital design methods incorporating modeling, simulation and hardware description languages like VHDL. It is constantly updated with the latest Xilinx development tools, which are used in the second half of the EE341/EE343 course. In combination with FPGA based evaluation boards the students have to implement logic designs based on schematics as well as with VHDL.

Beyond the education in HDL based digital design, the lab is used to teach digital signal processing, embedded systems, and modeling and simulation. All courses are supplemented with appropriate hardware. This lab is used for the class taught to CS students, EE 341, as well as the class taught to EE students, EE 343.

B. Computing Resources

LAB	HOURS
Bunnell Open Lab	Mon - Fri: 8 AM - 5 PM, except University holidays and periodic maintenance
Rasmuson Open Lab	Open 24 hours per day, 7 days per week
MBS Open Lab	Open 24 hours per day, 7 days per week
Rasmuson Training Labs	Scheduled through the Rasmuson Library media desk.
Chapman 103-Chapman Computer Lab	The hours for the Chapman Computer Lab change each semester based on availability of student employees. Presently they are: Mon - Fri: 8 AM – 9 PM, Sat - Sun: 10 AM – 5 PM Classes are held in this lab at various times M-F. Students can SSH remotely to any machine in Chapman Computer Lab.
Chapman 203 Lab	As access requires.
Chapman 205-Visualization Center	Open 24 hours for students in graphics-related computer science courses. Machines are available over the network via SSH.

Chapman 206-ASSERT Lab	Open to students in security-related computer science courses only.
Duckering 208-EE Digital Hardware Lab	As access requires.
Duckering 210-EE Digital Computational Lab	As access requires.

All users must comply with UAF's Acceptable Use Policy, available on the web at http://www.alaska.edu/oit/cito/OnlineResources.pdf.

The University of Alaska Fairbanks provides Ethernet access to students, faculty, and staff on campus. All dorm rooms have Ethernet access, as well as the majority of offices on campus. In addition, CTS offers wireless Ethernet in various locations on campus, as well as wired walk-up.

In addition to the labs on campus and the Ethernet hookups in residence halls and offices, wireless Ethernet and walk-up port access is available for students, faculty, and staff with laptops. There are two wireless networks available on campus.

In addition to on-campus network resources, UAF users can make use of the Virtual Private Network (VPN), a secure "private" connection between the client machine and the host network or system. This allows authorized users of UAF to access private servers or restricted services from off campus, home or when traveling.

UAF's Office of Information Technology provides an online learning experience through the Blackboard Course Management System, which provides a place for instructors to post course content, lectures, PowerPoint presentations, assignments, and more, making materials available for student access 24/7. In addition, Blackboard provides discussion boards and chat rooms where students and instructors can interact either synchronously (live) or asynchronously (delayed response). Blackboard is used both to supplement traditional on-campus classes, and also as a distance delivery tool, bringing the classroom to students' homes.

SOFTWARE: All UAF computers are site-licensed for Microsoft operating systems, Office, Visual Studio, Visio, and Project. A site license provides all UAF students, faculty, and staff with Symantec Anti-Virus for personal and work machines. The following licenses are maintained through OIT and managed through a key client application on each computer: KeyClient, KeyCheckout, Acrobat Pro, Citrix, Cyberduck, Dreamweaver, Fireworks, Flash Pro, Illustrator, InDesign, Photoshop Extended, Adobe Premiere Pro, CS4 Training Videos, Camtasia Studio, CopyIt, EndNote, LabView, Secure Shell, and VPN.

WEB RESOURCES: CTS provides faculty, staff, club and department webspace, email accounts and network storage. All students are provided with 1 GB of network storage that can be accessed anywhere on the UAF network.

TRAINING and PROFESSIONAL DEVELOPMENT: CTS provides training to students, staff and faculty on a variety of software applications including Google Apps, TOAD, Blackboard, Roxen, Elluminate Live (eLive!), Meeting Maker, MS Office products, and many others. A current schedule of training seminars is available at http://www.alaska.edu/oit/training/.

COMPUTING HELP RESOURCES: The Office of Information Technology provides a Support Center Help Desk that can be accessed by any UAF affiliate. The help desk provides general troubleshooting as well as informational handouts for configuring different system software and FAQs on different services provided. The helpdesk can be contacted by phone or through their website at http://www.alaska.edu/oit/sc/.

C. Guidance

LAB	SUPPORT AVAILABLE
Chapman 103-Chapman Computer Lab	In the event that a class is using the lab, the instructor is present, or student has arranged for a teaching assistant to be present for instructional assistance. The lab is staffed by CS graduate students who can guide students if needed.
Chapman 206-ASSERT Lab	There is an ASSERT Lab Manager available in an adjacent office during normal business hours. In addition, there is a graduate student available who is the assistant lab manager.

OIT Help Desk staff is located in Rasmuson Library Room 401, and is also available via email at helpdesk@alaska.edu and phone at 450-8300 (outside of Fairbanks 1-800-478-8226). Help Desk Hours (year-round): 7:30 AM to 9:30 PM, Mon-Fri; 10:00 AM to 6:00 PM, Sat-Sun.

While faculty members normally prefer to maintain their own desktop machines, and faculty provide significant input on the configuration and use of shared resources, the primary responsibility for maintaining shared hardware and software lies with CTS.

All on-line documentation for the software in the lab is installed whenever possible. In the event that on-line documentation is not available or inadequate, students or faculty can ask the lab assistant, who has access to any available hard-bound manuals, for help. No documentation is provided for the hardware facilities available in the lab, as students and faculty are discouraged from tampering with their configuration. Locks are installed on all of the workstations, and whenever possible, the configuration is locked on other devices such as printers. If a student or faculty member is having problems with a piece of hardware in the lab, they are expected to ask the lab assistant for help, rather than trying to fix it themselves.

CTS provides all network management, as well as assistance in the installation and maintenance of hardware and software for staff and faculty workstations. A staff member is employed half-time for the installation and maintenance of the Chapman Computer Lab, and hardware and software installation and maintenance on departmental machines.

In addition, a 0.5 FTE staff member is employed by the Computer Science Department for the installation and maintenance of the Chapman Computer Labs, and hardware and software installation and maintenance on departmental machines.

D. Maintenance and Upgrading of Facilities

The laboratory needs of the program are discussed annually in department faculty meetings.

Each workstation and server in the Chapman Computer Lab is replaced every three years using funds from the semester-based lab fee assessed to all students. The remaining hardware is replaced when it becomes obsolete, stops working, or as funds permit. Because the hardware and software needs of the faculty are diverse, requests for procurement and upgrades are largely made by individual faculty through the department chair. The ASSERT Lab equipment has always been maintained and upgraded through grants.

University-wide facilities and network upgrades are planned and supported by OIT. The Electrical Engineering lab is maintained by the EE department, with input from Computer Science faculty.

E. Library Services

The Rasmuson Library website, www.uaf.edu/library, provides detailed information about departments, services and collections within the Rasmuson Library, and also provides access to information through its online catalog (Goldmine). The entire system catalog can be searched by author, title, subject or other search method using library.uaf.edu/goldmine. This site can be accessed by students, faculty and the public. Current journal subscriptions and e-journals licensed for UAF use are available through the Journal List web page, also accessed through the Library web site.

Goldmine can be used to locate not only what is owned by the Rasmuson library but also what is owned by all the sites in the University of Alaska statewide system. For example, the Interlibrary Loan page provides information on how to obtain books, photocopies, or audiovisual materials that are not available on campus from other library locations, how long it takes, renewals, general policies, Web Document Delivery, how to access an online request form and more. In addition, a wide variety of subject-specific databases is available in the Library via the Elmernet local area network (only searchable in the library), and via the Internet to UAF students, faculty, and staff. On-campus users may access all resources listed on the Library website from any campus public, office, or dorm room computer. Off-Campus use of licensed e-resources (with the exception of the "Databases for Alaskans" collection) is restricted to UAF faculty, students, and staff, and requires that UAF users login using their UAF computer ID and password.

The library currently subscribes to more than 130 electronic databases including online indexes, full-text journal article collections, e-books and encyclopedias. Some of the database searches available to UAF students, faculty and staff are listed in Table 2. Additional resources are frequently added to the library web site, including article indexes & collections, alphabetical or subject lists for the most current listings, as well as access to information resources available online.

As part of UAF's core curriculum, undergraduate students must demonstrate their library proficiency either by completing LS101 (Library Information and Research) or by passing a competency exam. In LS101, students learn about library research using the Internet, and about finding information in a variety of subject areas.

Additionally, librarians and library staff are available to assist students in using library resources and can give guidance on how to best locate research and information resources both in the library and beyond, regardless of format. Assistance can be provided by phone, email and live chat with a librarian or library staff member.

The Rasmuson Library is the largest in the state, with more than 1.1 million volumes. Special collections include the world-class Alaska and Polar Regions collections, covering books, periodicals, archives, manuscripts, historical photographs, oral histories and maps. A branch of the Rasmuson Library, the Biosciences Library on West Ridge, contains a substantial collection of books and journals. The Geophysical Institute operates the Mather Library to support student, staff and faculty research needs in the geophysical area. Services provided by the Rasmuson Library include:

- Carrels - Available for grad students on a first-come/first-served basis.
- Circulation - Information about borrowing books & videos, overdue policies, and your library account.
- Conference and Meeting Rooms - reserve rooms within the library.
- Digital Photographic Services - professional digital imaging services available to the university community and to the public. Offers digital printing and high resolution scanning.
- Document Delivery - request book chapters and paper journal articles be scanned and emailed to you as a .pdf file.
- Interlibrary Loan - borrow material from other libraries.
- Instruction - whether your student, faculty, or staff, we can show you how to use our resources.
- Media Services - borrow media equipment such as digital cameras, camcorders, laptops, and more. Popular and reference DVDs and CDs are available for check out.
- Off-Campus Services - a unit set up to serve rural UAF students and faculty who do not have access to appropriate information resources in their town or village.
- Reference Services - need help with your research? We can help!
- Reserves - Reading materials for specified classes.
- Room Scheduling - Reserve rooms for study sessions, group meetings, conferences, and teaching.

The Rasmuson Collection Development Officer periodically polls all faculty on campus on program needs for books and/or subscriptions—here is her most recent email specifically to CEM:

Dear College of Engineering faculty and graduate students:

With Spring Semester almost over, I know most of you won't be thinking about library collections, but since we remain open and work through the summer months, it's a perfect time for us to acquire whatever books you might need for Fall Semester. We do have funds available for book purchases, and we are right now prioritizing our journal subscription requests as well, so if you have suggestions, please send them to me before you leave campus, if possible. I also welcome any assistance from faculty in weeding the older material out of our collections; we rely on your subject expertise to help us make these decisions.

I also wanted to fill you in on one of our latest acquisitions: the Earth and Environmental Sciences set of ebooks from Springer. These will be added to our Goldmine catalog shortly so that you can link directly, and will be hosted on the Springer platform which allows downloading to almost any device, as well as printing and simultaneous user access. This set has more than 1000 books, including multiple disciplines. If you want to glance through the title list here is the URL, choose Earth and Environmental:

http://www.springer.com/librarians/e-content/ebooks?SGWID=0-40791-12-377411-0

This book deal is good for all UA campuses; we hope to do more of these types of UA-statewide purchases in the future, so that faculty and students at all campuses can benefit.

Finally, it has been a banner year for use of the EBL or electronic books system. Use has more than tripled since we began this project several years ago. If you haven't tried EBL books yet, or you'd like me to demonstrate it for a group or individually, I'm happy to do so.

Please let me know if you have any questions, and feel free to stop by any time to share suggestions or concerns. Have a great summer!

Karen Jensen
Collection Development Officer
Rasmuson Library
University of Alaska Fairbanks

The electronic UAF library catalog is called Goldmine (library.uaf.edu/goldmine), and is an easy-to-use resource for searches. Electronic Books Online (EBL) provides both short-term loans and auto-purchasing options for ebooks, on all topics. Readers may view material online, download to a computer for a limited time, and copy or print a small amount of material from these ebooks. A login is required from both on and off campus. The library director noted the incredibly fast transition to electronic materials from traditional print materials: *"We circulated 34,572 physical books in 2010 while EBL in its first full year of use circulated 33,411 book titles. When combined with the library's other [electronic] book collections – Safari, Psycbooks, Netlibrary, Springer, Elsevier, etc. – we expect to see that the use of digital titles now substantially surpasses the circulation of more typical library materials."*

Available databases include:

Applied Science and Technology Abstracts FirstSearch Database	Engineering, mathematics, physics and computer technology.
ABI/INFORM Global ABI/INFORM Global Database	Indexing for articles in over 1,200 international business, management, and marketing journals, including many computer science representative trade journals. Beginning in 1970 with some full-text. Current search of "Computer Science" as subject yields 2,506 documents.
ACM Digital Library Core Package ACM Digital Library	Full text collection of every article published by ACM, including over 50 years of archives.
Computer Source EBSCOhost Database	Includes full-text and citations for current trends in high technology, covering topics such as computer science, programming, artificial intelligence, cybernetics, information systems, robotics, and software. Dates back to 1985.
Compendex® Engineering Village 2	Most comprehensive interdisciplinary engineering database in the world. Compendex contains over 8 million records and references over 5,000 international engineering sources including journal, conference, and trade publications. Coverage is from 1969 to present and the database is updated weekly.
IEEE All Societies Package IEEE Xplore	Includes access to abstracts and full-text of IEEE journals, transactions, and magazines published since 1998.
IEEE Core Proceedings IEEE Core Proceedings	Contains core collection of IEEE conferences from 1998 to present.

INSPEC ISI Web of Knowledge	Provides bibliographic information and some full-text from the world's leading scientific and technical literature, covering subjects such as physics, engineering, electronics, computers, and information technology. From 1969 to the present.
Institute of Physics (IOP) Online Journals IOP Journals Online	Access to a variety of full text journals in the area of physics, math, and engineering.
Kluwer Online springeronline.com	Full text journals, primarily in the sciences and social sciences, published by Kluwer Academic Publishers. Currently includes 96 Computer Science journals.
MathSciNet MathSciNet Database	Mathematical reviews on the Web. MathSciNet is a comprehensive database covering the world's mathematical literature since 1940.
NTIS CSA Internet Database Service	Produced by the National Technical Information Service, the NTIS database is the preeminent resource for accessing the latest U.S. government-sponsored research and worldwide scientific, technical, engineering, and business-related information.
Safari Tech Books Online safaribooksonline.com	An online library that provides full-text access to a current collection of 2,622 information technology books.
Science Citation Index Expanded and Web of Science Rasmuson Library-Article Indexes and Collections	Provides access to current and retrospective bibliographic information, author abstracts, and cited references found in the world's leading scholarly science and technical journals.
Science Direct Web Editions Sciencedirect.com	A current awareness service from Elsevier Publishing that provides full-text access to the current year journals in science, technology, and medicine that match UAF's print subscriptions.
Wiley Interscience Enhanced Access Wiley Interscience Home	Access to full text journals. Multidisciplinary coverage.

Overall, the library capabilities are quite adequate for the program.

At the end of fiscal year 2010 paper editions of book and serial backfiles were reported as 821,982 volumes; and microforms as 160,333 units. The current number of serial/journal subscriptions available in paper or microform is 1,763. Current serials/journals in digital format licensed for use by UAF students, faculty, researchers and staff exceed 50,000 titles. The Library also subscribes to many online indexes, full-text journal article collections, and encyclopedias.

In FY10, the library materials budget of $2,008,159 constituted 27% of the total operating budget of $7,402,601.

FY10 Books	$137,570
FY10 Periodicals	$371,927
FY10 Standing Orders *[includes some databases]*	$642,285
FY10 Electronic Databases *[incl. some aggregators of full-text journals]*	$817, 036
FY10 Video	$25,341
FY10 Binding	$14,000

Significant services provided by the library include access to online library catalogs, indexes, and full-text journals; a lending program for media & other electronic equipment for student & classroom use; the Interlibrary Loan program; and free document delivery services. In addition,

all open computing labs are available for use by UAF students, staff and faculty for library information retrieval as well as multiple other uses.

The Library seeks collection development suggestions from faculty, staff and students and determines adequacy of resources for campus programs proactively by cooperative collection building. The Library also actively participates in the review process for all new courses and programs. All faculty proposing changes or additions must contact the Library Collection Development Officer; together they determine library resource adequacy based on historical and current collecting profiles. No new course or program is approved if this review is negative, unless additional funding is identified to provide sufficient library resources.

Library Staffing

Computer Science library materials comprise part of the general book and journal library collections, and are not administered as a separate departmental library with a librarian supporting its exclusive needs. All library science faculty are assigned as liaisons to departmental groups of similar interests. At this time, the current Web Librarian serves as liaison to the Mathematics and Computer Science departments. Liaison librarians provide communication at the departmental level, monitor resource development, offer instruction and advice in the use of library collections and resources, and improve the library's resources in specific subject areas. The Computer Science Department also has support of the Library's collective team of librarians and staff who serve the entire campus, including the satellite Biosciences Library. The Library staff provides face-to-face assistance with library resources and services within the physical libraries, and delivers virtual information services to the desktops of UAF faculty, staff and students throughout the campus and the world.

In FY10, the total number of Library staff, excluding student assistants was 73.5, and includes the following staff categories: Executive, 1; Library Science Faculty, 12; APT (other professional staff), 13; and Classified (support personnel), 47.5. One additional campus librarian exclusively serves the Geophysical Institute and International Arctic Research Centers.

Although not affiliated with the library directly, the Office of Information Technology is also available to provide instructional and professional developmental services to local and rural campus faculty and staff, by utilizing technology such as: Smart Classrooms, Streaming Audio and Video, Satellite communication feeds, Interactive web applications, Internet based instruction and Video conferencing.

Technical Collection

The UAF library provides 1,309 printed Computer Science books, and provides access to more than 5,000 online books on computers and computer science, as well as numerous online videos on software training. These materials are available through the Goldmine library catalog. The libraries also subscribe to 497 Computer Science periodicals and trade journals (Table 1); the print titles are listed in Goldmine, and the online titles through our UAF Journals List. These sites can be accessed by students, faculty and the public. All students and faculty also have access to numerous online Computer Science resources, including the IEEE Explore Database, INSPEC, the ACM Digital Library core Package, through the library website: http://library.uaf.edu/onlinedatabases/ui/subject.php?id=12.

Supplementary texts are purchased each year through the Collection Development Department. For the FY10 fiscal year, the library allocated $3,000 to supplement the math & computer science collection. Additional requests are considered as needed. The amount allocated each fiscal year depends on the budget for that year. Due to vastly increased demand for online materials, the budget allocation for printed books has declined in recent years. However our current online journal and ebook access vastly surpasses any purchases we could have done for print materials in past years, with 10 times the number of ebooks now available over our annual purchase amount.

The Library encourages faculty and students to suggest books, journals and other materials for acquisition. Patrons may submit recommendations by email, phone, in person, or through the library's website form. This information is on the Library web site under Collection Development, along with the online Resource Purchase Suggestion Form.

Table 1 – UAF Rasmuson Library Subscription List for Computer Science

Title	PrintISSN	OnlineISSN	Coverage
2600	0749-3851		1984-present
3C ON-LINE	1078-2192		1994 - 1997
Accessibility and Computing, ACM SIGACCESS	1558-2337	1558-1187	2003 to present
Accessible Computing, ACM Transactions on	1936-7228	1936-7236	2008 to present
ACM SIGACT News	0163-5700		1969 to present
ACM SIGAda Ada Letters	1094-3641		1981 to present
ACM SIGAPL APL Quote Quad	0163-6006		1971 - 2007
ACM SIGART Bulletin	1053-4830		1970 - 2001
ACM SIGCSE Bulletin	0097-8418		1969 to present
ACM SIGFORTH Newsletter	1047-4544		1989 - 1994
ACM SIGICE Bulletin	1078-134X	1558-1144	1994 - 1997
ACM SIGITE Newsletter	1550-1469	1558-1071	2005 to present
ACM SIGMICRO Newsletter	1050-916X	1558-0296	1972 - 1992
ACM SIGNUM Newsletter	0163-5778		1966 - 1998
ACM SIGOA Newsletter	0737-819X		1980 - 1986
ACM SIGOIS Bulletin	0894-0819		1986 - 1996
ACM SIGPLAN Notices	0362-1340		1966 to present
ACM SIGSAC Review	0277-920X	1558-0261	1981 - 1997
ACM SIGSAM Bulletin	0163-5824		1967 to present
ACM SIGSMALL Newsletter	0272-720X		1978 - 1984
ACM SIGSMALL/PC Notes	0893-2875		1985 - 1993
ACM SIGUCCS Newsletter	0736-6892		1965 - 1998
ACM SIGWEB Newsletter	1931-1745	1931-1435	1992 to present
Acta Informatica		1432-0525	1996 to present
Affective Computing, IEEE Transactions on		1949-3045	2010 to present
AI and Society		1435-5655	1997 to present
AI Magazine	0738-4602		1997 to present
Algorithmica		1432-0541	1997 to present
Algorithms, ACM Transactions on	1549-6325	1549-6333	2005 to present
Annals of Software Engineering	1022-7091	1573-7489	1997 - 2002
Annals of the History of Computing, IEEE		1934-1547	1998 to present
Annual Simulation Symposium	1080-241X		1978 - 2007
Applied Artificial Intelligence		1087-6545	1996 to present
Applied Computing Review, ACM SIGAPP	1559-6915	1931-0161	1993 - 2002
Applied Intelligence		1573-7497	1997 to present
Applied Perception, ACM Transactions on	1544-3558	1544-3965	2004 to present
Applied Soft Computing		1872-9681	2001 to present
Architecture and Code Optimization, ACM Transactions on	1544-3566	1544-3973	2004 to present
AS/400 Systems Management	1086-881X		1997 - 1999
ASEE Prism	1056-8077		1996 to present
Asian Language Information Processing, ACM Transactions on	1530-0226	1558-3430	2002 to present
Asterisk Journal of Computer Documentation, ACM SIGDOC	0731-1001		1975 - 1999
Australian Journal of Educational Technology	1449-3098	1449-5554	1985 - 2003
Automated Software Engineering		1573-7535	1996 to present
Automatic Control and Computer Sciences		1558-108X	2007 to present
Autonomous and Adaptive Systems, ACM Transactions on	1556-4665	1556-4703	2006 to present
Behaviour and Information Technology	0144-929X	1362-3001	1996 to present
Byte	0360-5280		1975-1998
C O R S journal	0574-9638		1965 - 1970

Campus Wide Information Systems	1065-0741		1995 to present
Capacity Management Review CMR: A Monthly Report on Managing Computer Performance	1049-2194		1987 - 1998
CD Computing News	0893-4843		1997 to present
CD ROM Databases	0897-3296		1999 - 2000
Client Server News	1351-5500		1999 - 2007
Clinical data management	1073-6379		1999 - 2000
Cluster Computing		1573-7543	1998 to present
Communications in Computer Algebra, ACM	1932-2240		1967 to present
Communications of the ACM	0001-0782	1557-7317	1958 to present
Communications of the Association for Information Systems		1529-3181	2003 to present
Communications Technology	0884-2272		2005 - 2006
Computation Theory, ACM Transactions on	1942-3454	1942-3462	2009 to present
Computational Complexity		1420-8954	1991 to present
Computational Intelligence		1467-8640	1997 to present
Computational Logic, ACM Transactions on	1529-3785	1557-945X	2000 to present
Compute	0194-357X		1991 - 1994
Computer	0018-9162		1970 to present
Computer and Internet Lawyer	1531-4944		1997 to present
Computer Architecture Letters, IEEE		1556-6064	2002 to present
Computer Architecture News, ACM SIGARCH	0163-5964		1972 to present
Computer Business Review	1350-4665		1995 - 1998
Computer Communication Review, ACM SIGCOMM	0146-4833		1970 to present
Computer Fraud and Security	1873-7056		1996 to present
Computer Graphics and Applications, IEEE		1558-1756	1998 to present
Computer Graphics, ACM SIGGRAPH	0097-8930		1969 to present
Computer Journal		1460-2067	1999 to present
Computer Law and Security Review		1873-6734	1995 to present
Computer Lawyer	0742-1192		1999 - 2000
Computer Personnel, ACM SIGCPR	0160-2497		1964 - 2002
Computer Programming, DP and Related Services Industry Yearbook			1998 - 2002
Computer Protocols	0899-126X		1999 to present
Computer Science - Research and Development		1865-2042	2009 to present
Computer Science Education		1744-5175	1998 to present
Computer Security Update			2001 to present
Computer Software and Networking Industry Yearbook	0289-6540		1998 - 2002
Computer Systems, ACM Transactions on	0734-2071	1557-7333	1983 to present
Computer Vision, IET		1751-9640	2007 to present
Computer Weekly	0010-4787		2002 to present
Computer Workstations	0899-9783		1999 to present
Computer-Human Interaction, ACM Transactions on	1073-0516	1557-7325	1994 to present
Computers and Digital Techniques, IET		1751-861X	2007 to present
Computers and Society, ACM SIGCAS	0095-2737		1970 to present
Computers and the Physically Handicapped, ACM SIGCAPH	0163-5727		1971 - 2003
Computers in Entertainment	1544-3574	1544-3574	2003 - 2009
Computers, IEEE Transactions on		0018-9340	1998 to present
Computerworld	0010-4841		1987 to present
Computing and Visualization in Science		1433-0369	1997 to present
Computing Education, ACM Transactions on	1946-6626	1946-6226	2001 to present
Computing in Science and Engineering		1558-366X	1999 to present
Computing Japan	1340-7228		1997 - 1999
Computing Reviews	0010-4884	1530-6585	1969 - 2007
Computing Surveys, ACM	0360-0300	1557-7341	1969 to present
Conference on Computer and Communications Security	1543-7221		1993 - 2009
Conference on Human Factors and Computing Systems	1062-9432		1976 to present

Constraints		1572-9354	1997 to present
CPA Technology Advisor	1550-4743		2005 to present
Crossroads	1528-4972	1528-4980	1994 to present
Cryptography and Communications		1936-2455	2010 to present
Cybernetics and Systems Analysis	1060-0396	1573-8337	1997 to present
Cyberpsychology and Behavior : The Impact of the Internet, Multimedia and Virtual Reality On Behavior and Society		1557-8364	
Darwin	1536-2256		2002 - 2002
Data Base Management	1086-7575		2002 - 2003
Data Communications	0363-6399		1987 - 1999
Data Mining and Knowledge Discovery		1573-756X	1997 to present
Data Strategy	1745-1264		2008 to present
Database for Advances in Information Systems	1532-0936		1998 to present
Database Systems, ACM Transactions on	0362-5915	1557-4644	1976 to present
Database, ACM SIGMIS	0095-0033		1969 to present
Datamation (Online)	0011-6963		1998 - 1999
Dataquest	0970-034X		1996 - 2000
Dependable and Secure Computing, IEEE Transactions on		1545-5971	2004 to present
Design and Test of Computers, IEEE		1558-1918	1998 to present
Designs Codes and Cryptography		1573-7586	1996 to present
Discrete Mathematics and Theoretical Computer Science	1365-8050		2003 to present
Distributed and Parallel Databases		1573-7578	1997 to present
Distributed Computing		1432-0452	1997 to present
Distributed Systems Online, IEEE		1541-4922	2000 to present
Dr. Dobb's Journal : Software Tools for the Professional Programmer	1044-789X		1997 - 2009
Dynamics and Stability of Systems		1465-3389	1999 - 2001
Economic Computation and Economic Cybernetics Studies and Research	0424-267X	1842-3264	2009 to present
Education and Information Technologies		1573-7608	1997 to present
Edutech Report	0883-1327		2004 - 2006
Electronic Commerce News	1086-2870		1996 - 2004
Electronic Letters on Computer Vision and Image Analysis	1577-5097	1577-5097	2002 to present
Embedded Systems Programming			1999
Empirical Software Engineering		1573-7616	1996 to present
Engineering with Computers		1435-5663	1997 to present
Ethics and Information Technology		1572-8439	1999 to present
Evolutionary Computation	1063-6560	1530-9304	1993 to present
Evolutionary Computation, IEEE Transactions on	1089-778X		1998 to present
Eweek	1530-6283		2000 to present
Expert Magazine, IEEE	0885-9000		1986 - 1997
Expert Systems		1468-0394	1998 to present
Expert Systems with Applications		1873-6793	1995 to present
Explorations Newsletter, ACM SIGKDD	1931-0145	1931-0153	1999 to present
ExtremeTech com	1551-8167		2002 to present
Formal Aspects of Computing		1433-299X	1997 to present
Formal Methods in Software Practice	1533-9955		1998 - 2000
Formal Methods in System Design		1572-8102	1997 to present
Fortran Forum	1061-7264	1931-1311	1982 to present
Fractals		1873-2887	1995 to present
Frontiers of Computer Science in China		1673-7466	2007 to present
Fuzzy Systems, IEEE Transactions on	1063-6706		1998 to present
Game developer	1073-922X		1998 to present
Genetic Programming and Evolvable Machines		1573-7632	2000 to present
Hewlett Packard Journal	0018-1153		1995 - 1998
Higher-Order and Symbolic Computation		1573-0557	1998 - 2009
History and Computing	0957-0144		1998 - 2002

Home Office Computing	0899-7373		1995
Human-Computer Interaction		1532-7051	1985 to present
IBM Journal of Research and Development	0018-8646		1997 to present
IBM Systems Journal	0018-8670		1987 to present
Industrial Informatics, IEEE Transactions on	1551-3203		2005 to present
Info World Canada	1208-4182		1997
Informatik - Forschung und Entwicklung	0178-3564		1997 - 2008
Informatik Spektrum		1432-122X	1997 to present
Information and System Security, ACM Transactions on	1094-9224	1557-7406	1998 to present
Information Forensics and Security, IEEE Transactions on		1556-6021	2006 to present
Information Intelligence Online Newsletter	0194-0694		1996 - 2002
Information Management and Computer Security	0968-5227		1995 to present
Information Retrieval		1573-7659	1999 to present
Information Systems Research		1526-5536	1990 to present
Information Systems Security	1065-898X		1995 - 2007
Information Technologies and International Development		544-7537	2003 to present
Information Technology in Biomedicine, IEEE Transactions on		1558-0032	1998 to present
Information Theory, IEEE Transactions on		1557-9654	1998 to present
Information Theory, IRE Professional Group on			1953 - 1954
Information Theory, IRE Transactions on	0096-1000		1955 - 1962
Information Today	8755-6286		1987 to present
Information Visualization		1473-8724	2002 to present
InformationWeek	8750-6874		1994 to present
INFORMS Journal on Computing		1526-5528	1996 to present
InfoStor	1097-2501		1999 - 2007
Infotech Update	0308-9487		1992 - 2007
InfoWorld	0199-6649		1987 - 2007
Innovations in Systems and Software Engineering		1614-5054	2005 to present
Inroads, ACM	2153-2184		2010 to present
Installation Management Review, ACM SIGCSIM	0163-5972		1969 - 1977
Intelligent Automation and Soft Computing	1079-8587		2006 to present
Intelligent Enterprise	1524-3621		1999 - present
Intelligent Systems and their Applications, IEEE	1094-7167		1998 - 2000
Intelligent Systems, IEEE	1541-1672		2001 to present
Interacting with Computers	0953-5438	1873-7951	1989 to present
Interactions	1072-5520	1558-3449	1994 to present
Interactive Age	1080-4927		1995 - 1995
Interactive Daily	1083-141X		1995 - 1997
Interactive Week	1078-7259		1998 - 2001
International Conference on Autonomous Agents	1534-4797		1997 - 2009
International Conference on Knowledge Capture	1549-5922		2001 - 2009
International Journal of Computer Vision		1573-1405	1996 to present
International Journal of Computers and Applications	1206-212X		2006 to present
International Journal of Cooperative Information Systems	0218-8430		1999 to present
International Journal of Expert Systems	0894-9077		1996 - 1997
International Journal of Foundations of Computer Science		1793-6373	1999 to present
International Journal of High Speed Computing	0129-0533		1999 - 2004
International Journal of Human Computer Interaction		1532-7590	1997 to present
International Journal of Information Security		1615-5270	2001 to present
International Journal of Management and Innovation	2070-8521		2009 to present
International Journal of Parallel Programming		1573-7640	1997 to present
International Journal of Shape Modeling		1793-639X	2000 to present
International Journal of Software Engineering and Knowledge Engineering		1793-6403	1999 to present
International Journal of Systems Science		1464-5319	1999 to present
International Journal of Wireless Information Networks		1572-8129	1997 - 2009

International Journal on Document Analysis and Recognition		1433-2825	1998 to present
International Journal on Software Tools for Technology Transfer		1433-2787	1997 to present
International Review of Law, Computers and Technology		1364-6885	1996 to present
International Software Process Workshop	1097-1521		1988 - 1996
International Transactions in Operational Research		1475-3995	1994 to present
International Workshop on System-Level Interconnect Prediction	1544-5623		2000 to present
Internet Business	1414-9117		2002
Internet Computing, IEEE	1089-7801		1998 to present
Internet Technology, ACM Transactions on		1557-6051	2001 to present
IT architect	1557-2145		1998 - 2006
IT Professional	1520-9202		1999 to present
IT Training	0954-7940		2001 - 2007
Japan Inc	1345-4846		1999 - 2009
Journal in Computer Virology		1772-9904	2005 to present
Journal of Automated Reasoning		1573-0670	1996 to present
Journal of Computational Analysis and Applications		1572-9206	1999 - 2003
Journal of Computational and Graphical Statistics	1061-8600	1537-2715	1992 - 2004
Journal of Computer Documentation, ACM	1527-6805	1557-9441	2000 - 2002
Journal of Computer Information Systems	0887-4417		1997 to present
Journal of Computer Science and Technology		1860-4749	1997 to present
Journal of Computer Security		1875-8924	1996 to present
Journal of Computers in Mathematics and Science Teaching	0731-9258		1996 to present
Journal of Computing and Information Science in Engineering	1530-9827		2001 to present
Journal of Computing Sciences in Colleges	1937-4771	1937-4763	1991 to present
Journal of Cryptology		1432-1378	1997 to present
Journal of Database Management	1533-8010		2000 - 2009
Journal of Educational Computing Research		1541-4140	2003 to present
Journal of Experimental Algorithmics, ACM	1084-6654	1084-6654	1996 to present
Journal of Grid Computing		1572-9184	2003 to present
Journal of High Speed Networks		1875-8940	1996 to present
Journal of Information Privacy and Security	1553-6548		2005 to present
Journal of Information Systems Education	1055-3096		2000 to present
Journal of Intelligent Information Systems		1573-7675	1996 to present
Journal of Interconnection Networks		1793-6713	2000 to present
Journal of Internet Services and Applications		1869-0238	2010 to present
Journal of Machine Learning Research		1533-7928	2000 to present
Journal of Network and Systems Management		1573-7705	1997 to present
Journal of Organizational and End User Computing	1546-5012		2000 - 2009
Journal of Parallel and Distributed Computing		1096-0848	1995 to present
Journal of Real-Time Image Processing		1861-8219	2006 to present
Journal of Research on Computing in Education	0888-6504		1990 - 2001
Journal of Scientific Computing		1573-7691	1997 to present
Journal of Supercomputing		1573-0484	1996 to present
Journal of Systems Integration		1573-8787	1997 - 2001
Journal of the ACM (JACM)	0004-5411	1557-735X	1954 to present
Journal of the Brazilian Computer Society	0104-6500	1678-4804	2010 to present
Journal on Computing and Cultural Heritage	1556-4673	1556-4711	2008 to present
Journal on Computing, SIAM		1095-7111	1997 to present
Journal on Data and Information Quality, ACM	1936-1955	1936-1963	2009 to present
Journal on Educational Resources in Computing		1531-4278	2001 - 2009
Journal on Emerging Technologies in Computing Systems, ACM	1550-4832	1550-4840	2005 to present
Journal on Scientific Computing, SIAM		1095-7197	1997 to present
Knowledge and Data Engineering, IEEE Transactions on		1558-2191	1998 to present
Knowledge and Information Systems		0219-3116	2000 to present
Knowledge Discovery from Data, ACM Transactions on	1556-4681	1556-472X	2007 to present

Knowledge Management (Oxford)	1463-1822		2000 - 2002
Knowledge-Based Systems	0950-7051	1872-7409	1987 to present
LAN Product News			1999 to present
Letters on Programming Languages and Systems , ACM	1057-4514	1557-7384	1992 - 1993
Lifetime Data Analysis		1572-9249	1996 to present
Linux Journal	1075-3583		1994 to present
LISP and Symbolic Computation	0892-4635		1997 - 1998
Lisp Pointers, ACM SIGPLAN	1045-3563		1987 - 1995
Machine Learning		1573-0565	1986 to present
Machine Vision and Applications		1432-1769	1997 to present
MacWorld : The Macintosh Magazine	0741-8647		1989 to present
Mainframe Computing			1999 to present
Managed Network Services News	1092-292X		1998 - 1998
Markup Languages Theory and Practice		1537-2626	1999 - 2001
Mathematical and Computer Modelling of Dynamical Systems		1744-5051	1998 to present
Mathematics in Computer Science	1661-8270	1661-8289	2007 to present
Methodology And Computing In Applied Probability		1573-7713	1999 to present
Micro, IEEE	0272-1732		1998 to present
Micromath	0267-5501		2002 - 2005
Minds and Machines		1572-8641	1996 to present
Mobile Computing and Communications Review, ACM SIGMOBILE	1559-1662	1931-1222	1997 to present
Mobile Computing, IEEE Transactions on	1536-1233		2002 - present
Mobile Networks and Applications	1383-469X	1572-8153	1996 to present
Modeling and Computer Simulation, ACM Transactions on	1049-3301	1558-1195	1991 to present
Multibody System Dynamics		1573-272X	1997 to present
MultiMedia and Internet Schools	1546-4636		1998 to present
Multimedia Systems		1432-1882	1997 to present
Multimedia Tools and Applications		1573-7721	1996 to present
Multimedia, IEEE	1070-986X		1998 to present
Multimedia, IEEE Transactions on	1520-9210		1999 to present
Natural Computing		1572-9796	2002 to present
Network and Service Management, IEEE Transactions on		1932-4537	2004 to present
Network Magazine	1539-8137		2002 - 2005
Network World	0887-7661		1987 to present
Network, IEEE		1558-156X	1998 to present
Networking, IEEE/ACM Transactions on		1063-6692	1993 to present
Neural Computing and Applications		1433-3058	1997 to present
New Paradigms in Information Visualization and Manipulation	1544-8118		1997 - 1999
OEM Magazine	1071-8990		1997 - 1997
ON Magazine	1533-4090		2001-2002
Online Product News			1997 to present
Online Reporter	1364-7113		1999 - 2007
Online User	1085-7257		1996 - 1997
OOPS Messenger, ACM SIGPLAN	1055-6400		1990 - 1996
Operating Systems Review, ACM SIGOPS	0163-5980		1969 to present
Optical Memory and Neural Networks		1934-7898	2007 to present
Optical Networks Magazine		1572-8161	2000 - 2003
Parallel Algorithms and Applications	1063-7192		2002 - 2004
Parallel and Distributed Systems, IEEE Transactions on		1558-2183	1998 to present
Parallel and Distributed Technology: Systems and Applications, IEEE	1063-6552		1993 - 1996
Parallel and Large-Data Visualization and Graphics	1543-4346		1999 - 2003
Parallel Processing Developments			1996
Parallel Processing Letters		1793-642X	1999 to present
Parallel Programming and JAVA			1998
Pattern Analysis and Applications		1433-755X	1998 to present

Pattern Analysis and Machine Intelligence, IEEE Transactions on	0162-8828		1998 to present
Pattern Recognition and Image Analysis		1555-6212	2006 to present
PC Business Products			1999 to present
PC Computing		1081-8642	1998 - 2000
PC Magazine	0888-8507		1999 to present
PC Week	0740-1604		1998 - 2005
PC World	0737-8939		1994 to present
PC World.Com	0737-8939		1997 - 2006
Performance Computing	1529-3963		1998 - 2000
Performance Evaluation Review, ACM SIGMETRICS	0163-5999		1972 to present
Personal and Ubiquitous Computing		1617-4917	1997 to present
Personal Computing	0192-5490		1987 - 1990
Pervasive Computing, IEEE		1558-2590	2002 to present
Photonic Network Communications		1572-8188	1999 to present
Principles and Practice of Parallel Programming	1542-0205		1990 to present
Problems of Information Transmission	0032-9460	1608-3253	2001 to present
Productivity Software	1040-1482		1997 to present
Programming and Computer Software		1608-3261	2000 to present
Programming Languages and Systems, ACM Transactions on	0164-0925	1558-4593	1979 to present
Quantum Information Processing		1573-1332	2002 to present
Queue, ACM	1542-7730	1542-7749	2003 to present
Real Time Systems		1573-1383	1996 to present
Reconfigurable Technology and Systems, ACM Transactions on	1936-7406	1936-7414	2008 to present
Record, ACM SIGMOD	0163-5808		1969 to present
Records, ACM SIGMultimedia		1947-4598	2009 - 2009
Red Herring	1080-076X		2000 to present
Reliable Computing		1573-1340	1997 - 2007
Requirements Engineering		1432-010X	1997 to present
Science in China Series F: Information Sciences		1862-2836	2008
Science of Computer Programming		1872-7964	1995 to present
Scientific and Technical Information Processing		1934-8118	2007 to present
Scientific Computing and Instrumentation	1524-2560		2005
Scientific Programming		1875-919X	1997 to present
Scientometrics		1588-2861	1997 to present
Security and Privacy, IEEE		1558-4046	2003 - present
Security Systems News	1528-0519		2001 to present
Security Technology Executive	1946-8474		2005 to present
SIAM Review		1095-7200	1959 to present
SIGEVOlution, ACM		1931-8499	2006 to present
SIGLASH newsletter	0036-147X	1931-1095	1976 - 1981
SIGMAP Bulletin	0163-5786	1931-1184	1969 - 1983
SIGMINI newsletter	0163-576X		1975 - 1978
SIGPC Notes	0163-5816		1978 - 1982
SIGSPATIAL Special	1946-7729		2009 to present
Simulation Digest, ACM SIGSIM	0163-6103		1971 - 1998
Soft Computing		1433-7479	1997 to present
Software and Systems Modeling		1619-1374	2002 to present
Software Concepts and Tools		1432-2188	1998 - 2000
Software Development Times	1528-1965		2003 - 2007
Software Engineering and Methodology, ACM Transactions on	1049-331X	1557-7392	1992 to present
Software Engineering Notes, ACM SIGSOFT	0163-5948		1976 to present
Software Engineering, IEEE Transactions on	0098-5589		1976 to present
Software Magazine	0897-8085		1998
Software Quality Journal		1573-1367	1997 to present
Software, IEEE	0740-7459		1984 to present

Software, IET		1751-8814	2007 to present
Statistics and Computing		1573-1375	1997 to present
Storage, ACM Transactions on	1553-3077	1553-3093	2005 to present
System Development Management	1096-7893		2002 - 2003
System Dynamics Review		1099-1727	1996 to present
Systems Analysis Modelling Simulation	0232-9298		2002 - 2003
Systems Journal, IEEE	1932-8184		2007 to present
Systems, Man and Cybernetics, IEEE Transactions on	0018-9472		1971 - 1995
Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on	1083-4427		1998 to present
Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on	1083-4419		1998 to present
Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on		1558-2442	1998 to present
Technology and Society Magazine, IEEE	0278-0097		1998 to present
Technology Meetings	1521-5202		2002 - 2002
Technometrics	0040-1706	1537-2723	1959 - present
Theory of Computing Systems		1433-0490	1997 to present
UAE Information Technology Report	1750-5178		2009 to present
Universal Access in the Information Society		1615-5297	2001 to present
UNIX Update			1999 to present
User Modeling and User Adapted Interaction		1573-1391	1995 to present
Very Large Scale Integration Systems, IEEE Transactions on		1557-9999	1998 to present
Virtual Reality		1434-9957	1997 to present
Virtual Reality Modeling Language Symposium	1534-939X		1995 - 2002
VIS: IEEE Visualization	1070-2385		1990 - 2004
Visual Computer, The		1432-2315	1997 to present
Visualization and Computer Graphics, IEEE Transactions on		1941-0506	1998 to present
VLDB Journal, The		0949-877X	1992 to present
Web, ACM Transactions on the	1559-1131	1559-114X	2007 to present
Windows Magazine	1060-1066		1994 - 1999
Wired	1059-1028		1995-1999
Wireless Business Forecast	1069-3416		2004 - 2007
Wireless Communications, IEEE	1536-1284		2002 to present
Wireless Communications, IEEE Transactions on		1558-2248	2002 to present
Wireless Networks		1572-8196	1997 to present
World Wide Web		1573-1413	1998 to present
Worldwide Computer Products News	1363-9889		1999 to present
Year 2000 Practitioner	1096-4479		1999 - 1999

F. Overall Comments on Facilities

Describe how the program ensures the facilities, tools and equipment used in the program are safe for their intended purposes (See the 2011-2012 APPM II.G.6.b.(1)).

Faculty and staff monitor the safety of facilities, tools and equipment used to deliver the program. Additionally, the UAF Office of Environmental Health Safety and Risk Management and the Provost's Office have safety standards that must be followed for all university facilities and processes. One of the CEM technicians, Paul Brown, is the college safety officer and examines CEM labs, facilities and processes to ensure safety.

CRITERION 8. INSTITUTIONAL SUPPORT

A. Leadership

The department chair is an elected 2-year position. Faculty Senate policy (www.uaf.edu/uafgov/faculty-senate/policies-procedures/department-chair-policy) defines the role of the department chair as:

a. The department chair is the administrative and academic officer of the department and as such has the primary responsibility and authority for: (1) leadership in developing high quality academic programs which fulfill department, college, and university objectives; (2) leadership in the implementation of college and university policies and programs at the department level; (3) leadership in developing resource requests and an appropriate departmental budget; and (4) service on the college/school executive committee.
b. The department chair is first a faculty member. The department chair is primarily a teacher-scholar serving as a leader of his/her department colleagues. The department chair is a role model for faculty responsibility.
c. The department chair is responsible for providing mechanisms and processes for members' participation in discussion and decision making within the department. All members of the department should be informed of these mechanisms and processes. Regular meetings should be held for purposes of communicating information, discussing issues, and making decisions on department matters.
d. The department chair is expected to communicate faculty perspectives and concerns to the administration and other segments of the community as appropriate. The department chair is the primary spokesperson the faculty of the department. The department chair will also convey administration views and concerns to the faculty.

B. Program Budget and Financial Support

The dean determines the budget for the departments, given the funding allocated to the college. Starting with historical data, continuation budgets are developed for each department by adding budget increments, if applicable, to the budget levels from the previous fiscal year. Travel and equipment categories are not initially funded. Central funding for travel is distributed later to departments and awarded competitively to faculty from the CEM and INE travel programs. The equipment budget is funded centrally through a number of sources. These funds are distributed to departments toward the end of the fiscal year. Any equipment costing less than $5K is classified as a commodity. Student fees in computer labs and other labs are directed to the appropriate department and reinvested in the laboratories.

The Computer Science Department has assessed a $75 per student fee for students enrolled in CS courses and other students who require the use of the lab for certain courses in mathematics and statistics since 2004. Originally set at $42, the fee is currently $75. This fee provides adequate income (approximately $30,000 - $40,000 per academic year) for replacement and upgrade of the computers (generally on a 3-year cycle), software purchases and licenses as well as some assistance with staffing the lab. During Summer 2011, the lab is being completely refurbished

and new machines installed throughout. Graduate teaching assistants and undergraduate lab attendants have assigned times for monitoring the lab and provide seven days/week staffing of the lab during the academic year.

Finally, the department has a track record of securing grant funding to build major new investments in teaching and research infrastructure. UAF has an internal competitive process via the Technology Advisory Board (TAB), which allocates approximately $120K per semester for classroom technology, both large and small. TAB funds have supported the ASSERT computer security lab, and embedded programming hardware used in computer architecture courses. We have also obtained external funding from several sources, which have supported the ASSERT lab, the CS Bioinformatics Powerwall, and the NetRun web-accessible student programming environment.

1. **Describe how teaching is supported by the institution in terms of graders, teaching assistants, teaching workshops, etc.**

   The CS budget includes funds for three to four graduate teaching assistants for the CS department, which are used for grading, assisting in the lab, and helping students in higher enrollment courses. UAF has an active faculty development office, and regularly sponsors teaching workshops and guest lectures on teaching methods.

2. **To the extent not described above, describe how resources are provided to acquire, maintain and upgrade the infrastructures, facilities and equipment used in the program.**

   The CS department has budget for discretionary purchases ($28,200 of which is annually distributed from an initiative fund resulting from a CS proposal). CS priorities are discussed each year along with external opportunities to fund each. CS budget is used to fund those that are not likely to be funded through other avenues. Specific CS department infrastructure, such as our student accessible version control software server, is also maintained with department funds. General software infrastructure, such as Banner or Blackboard, as well as network infrastructure such as wiring and wireless access points, is provided and maintained centrally by the UAF Office of Information Technology, which has been generally responsive to our requests.

3. **Assess the adequacy of the resources described in this section with respect to the students in the program being able to attain the student outcomes.**

   Generally, our faculty and equipment support are sufficient for students to take core and elective courses on a reasonable schedule, perform hands-on experiments with cutting-edge computer hardware and software both inside and outside the classroom, and finish their degree familiar with modern computer science. There are some challenges that result directly from in the growth of graduate programs, which puts additional pressure on faculty.

C. Staffing

In addition to department staffing, CEM has the following positions that are shared by all the engineering programs for a variety of direct and indirect instructional support:

- Two network technicians
- One technician and building safety officer
- One mechanical technician
- One academic advisor
- One recruiting coordinator
- One chief fiscal officer
- One academic manager

The Institute of Northern Engineering is part of CEM and has several dedicated technicians and administrative personnel, which can also be used on an "as needed" basis.

The Department of Computer Science employs one full-time secretary. One full-time staff member works with the CS department at the grant level, serving as administrative support for funded grants of the department. A half-time computer lab manager is employed in the Computer Science student lab. Seven full-time faculty and other part-time faculty and staff are served by this administrative staff.

Staff participate regularly in training sessions offered by the university in computers and HR, travel, purchasing, payroll and other issues as appropriate.

D. Faculty Hiring and Retention

The process of hiring a new faculty starts with a memo from the department chair requesting permission from the Provost (through the Dean) to hire. Once the permission is received, a committee is setup by the program faculty in consultation with the department chair and Dean. The committee develops the job description and follows university guidelines in the hiring process. Faculty searches, an integral part of the hiring process, are typically international.

As soon as the formalities are completed, the job is posted on UAKJobs and formal ads are posted at the ACM website as well as other targeted venues depending on the identified specialty needs identified in the search. After achieving a certain pool of applicants or after a specified date (determined by the search committee) the candidates go through multiple screening stages such as a review of the resume and qualifications, a telephone interview, and on campus visits and reference checks. Near the end of the process, the committee makes a recommendation to the Dean in the form of a ranked list. Once the Dean's selection is made and approved by the Provost, an offer is made. Offered salaries typically conform to the Oklahoma State University salary survey.

Retention strategies for new faculty include targeted start up funds to enable the new faculty to develop a successful research program early in their UAF career. Additionally, lower teaching loads are offered in the first two years, along with reduced service workload. All new faculty are assigned, or may choose, a faculty mentor, typically in their department, to help assist with the

transition to a demanding academic career. Faculty development opportunities, through CEM & INE travel grants and through the UAF Office of Faculty Development, are intended to help with retention. If a current faculty member has a formal or informal job offer from another employer, the Dean has the option of increasing the salary for the faculty member in the form of a "retention raise."

E. Support of Faculty Professional Development

Sabbaticals are governed by the Collective Bargaining Agreement between UAF and the faculty union. Tenured or tenure track unit members who have completed at least 5 consecutive years of service within the unit are eligible for consideration to take sabbatical leave during the 6th or subsequent year of service. However, faculty consult with the department chair prior to applying for sabbatical leave in order to help the department and the program plan for the absence. An application for sabbatical leave is ranked by the department chair, the CEM Peer Review Committee, the CEM Dean and awarded by the UAF Provost. Sabbatical leaves are granted for periods of one academic year at the rate of six months' salary or one semester at the rate of one semester's salary.

Professional development of faculty has been discussed in Criterion 6-D. As mentioned, faculty sponsor their professional development activities through a combination of competitively awarded CEM academic travel grants, external research grants, and sometimes through non-university sponsors. The college provides additional travel funding to the departments, and faculty course buyouts have allowed the department to fund travel internally. Faculty have been able to undertake professional development activities, and have remained current, however there is no guaranteed funding to the college for this purpose.

Our faculty members are encouraged to and regularly engage in professional activities outside the University, and are recognized internationally for their work in Computer Science.

The Office of Faculty Development provides professional development opportunities for all faculty in the areas of teaching, learning, and scholarship. The Office provides assistance with travel, mentoring, promotion & tenure, and instructional technology (through the Center for Academic Technology). In addition, the Office also maintains a collection of resource materials on these topics in the office and at the Rasmuson Library. Regular workshops, panel discussions, and seminars are held throughout the year. Although these are specifically designed for new faculty, they are open to all faculty, and can be audio-conferenced to the rural campuses if needed.

PROGRAM CRITERIA FOR COMPUTER SCIENCE PROGRAMS

3j) The program must enable students to attain, by the time of graduation, an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the

tradeoffs involved in design choices. This is shown in the table in section 4.B, which lists the performance criteria for the j) student outcome that are developed in the 2nd and 3rd year courses.

3k) The program must enable students, by the time of graduation, an ability to apply design and development principles in the construction of software systems of varying complexity. This is also show in the table in section 4.B, which are developed throughout the curriculum but measured and evaluated in CS 471.

5.a.1) Coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture is met by the required courses CS 301, CS 311, CS 331, and either CS 441 or EE 341.

5.a.2) An exposure to a variety of programming languages and systems is met by CS 311, which covers a variety of programming languages on various systems.

5.a.3) Proficiency in at least one higher-level language is met by the required CS 201, CS 202 and CS 311 sequence, which all use C++ as the implementation language.

5.a.4) Advanced course work that builds on the fundamental course work to provide depth is met by all of our 400-level CS electives. Though not required, students may also choose a focus area of Information Assurance or Computer Graphics.

5.b.1) Mathematics is met by the required courses MATH 307 (Discrete Math) and STAT 300.

5.b.2) The Science component is met by the required calculus-based physics sequence PHYS 211 and PHYS 212.

APPENDICES

Appendix A – Student Outcomes Rubrics

Appendix B – Course Syllabi

Appendix C – Equipment

Appendix D – Institutional Summary

Appendix E – Faculty Vitae

Appendix F – Table of Substitutions

Appendix G – Advising Workshop Notes

Appendix A – Student Outcomes Rubrics

Student Outcomes with Rubric and Tripwire

		Beginning (1)	Developing (2)	Accomplished (3)	Exemplary (4)	Tripwire
a)	**An ability to apply knowledge of computing and mathematics appropriate to the discipline**					
1	Ability to select the proper data structure to solve a problem	Selected a data structure that is inappropriate (likely one the student is most familiar with)	Selected a data structure that will work, but is not optimal and requires more code to implement	Selected the optimal data structure from the basic types	Designed a hybrid data structure that was superior to the basic types	< 1.5
		Selected a data structure that is inappropriate (likely one the student is most familiar with)	Selected a data structure that will work, but is not optimal and requires more code to implement	Selected the optimal data structure from the basic types	Designed a hybrid data structure that was superior to the basic types	< 2.9
2	Ability to determine the efficiency class of an iterative algorithm	Can set up the summation for an iterative algorithm	Can only derive simple summations	Correctly derives most summations	Correctly derives difficult summations	< 2.9
3	Ability to determine the efficiency class of a recursive algorithm	Can set up the recurrence relation for a recursive algorithm	Can set up and solve simple recurrence relations	Can set up and solve most recurrence relations	Can set up and solve difficult recurrence relations	< 2.9
b)	**An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution**					

1	Ability to recognize the complexity class of a problem	Determines the wrong complexity class	Determines the correct complexity class by guessing or w/o proper justification	Determines correct complexity class based on best solution they can derive	Determines correct complexity class and provides justification (e.g. decision trees)	< 2.9
2	Ability to use abstraction to solve a given problem with an existing algorithm	Fails to solve problem correctly	Solves problem from scratch, no apparent attempt to use existing algorithms	Solves problem using existing algorithms as appropriate, solution is close to optimal	Solves problem using existing algorithms producing the optimal and elegant solution	< 2.9
3	Ability to create a software design document.	Requirements are incomplete, unorganized and ambiguous.	50% of the requirements are incomplete and/or ambiguous	20% of the requirements are incomplete and/or ambiguous	Less than 5% of the requirements are incomplete and/or ambiguous	< 2.9

c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

1	Ability to design an efficient algorithm to solve a problem (Use same/similar problem for all levels, but should allow improvement as students gain mastery of CS)	Algorithm fails on non-standard inputs and is lacking in elegance and efficiency	Algorithm works on standard inputs but is lacking in elegance or efficiency	Algorithm is optimal efficiency class, but may not be elegant or handle all input	Algorithm is optimal efficiency class, elegant and handles all input	<1.9
		Algorithm fails on non-standard inputs and is lacking in elegance and efficiency	Algorithm works on standard inputs but is lacking in elegance or efficiency	Algorithm is optimal efficiency class, but may not be elegant or handle all input	Algorithm is optimal efficiency class, elegant and handles all input	<2.4

		Algorithm fails on non-standard inputs and is lacking in elegance and efficiency	Algorithm works on standard inputs but is lacking in elegance or efficiency	Algorithm is optimal efficiency class, but may not be elegant or handle all input	Algorithm is optimal efficiency class, elegant and handles all input	<2.9
2	Ability to measure actual performance on a given architecture	Unable to determine performance (e.g. "slow")	Determines performance by manually timing (e.g. stopwatch)	Accurate, automated and repeatable measurement of performance	Accurate, automated and repeatable measurement with some statistical modeling (e.g. confidence interval)	< 2.9
3	Ability to design a software system based solely on a Software Requirements Document	Most design choices are wrong (e.g. which DB to use)	Design choices mostly correct, but no documented rationale for each choice	Design choices are correct, but little rationale is documented or is an afterthought	Design choices are well thought out with rationale for each documented	< 3.5
4	Ability to implement a software system	Only partial functionality is delivered	Most functionality is delivered, but program crashes too often	All important functionality is delivered with infrequent crashes	Software is "professional quality"	< 2.9

d) An ability to function effectively on teams to accomplish a common goal

1	Ability to create a software requirements document for a real-world client	Requirements are incomplete, unorganized and ambiguous.	50% of the requirements are incomplete and/or ambiguous	20% of the requirements are incomplete and/or ambiguous	Less than 5% of the requirements are incomplete and/or ambiguous	< 2.9
		Requirements are incomplete, unorganized and	50% of the requirements are incomplete and/or	20% of the requirements are incomplete and/or	Less than 5% of the requirements are incomplete	< 2.9

		ambiguous.	ambiguous	ambiguous	and/or ambiguous	
2	Ability to design a large software system for a real-world client	Most design choices are wrong (e.g. which DB to use)	Design choices mostly correct, but by accident	Design choices are correct, but no rationale documented	Design choices are well thought out with rationale for each documented	< 2.9
		Most design choices are wrong (e.g. which DB to use)	Design choices mostly correct, but by accident	Design choices are correct, but no rationale documented	Design choices are well thought out with rationale for each documented	< 2.9
3	Ability to implement and deliver a large software system to a real-world client	Only partial functionality is delivered	Most functionality is delivered, but program crashes too often	All important functionality is delivered with infrequent crashes	Software is "professional quality"	< 2.9
		Only partial functionality is delivered	Most functionality is delivered, but program crashes too often	All important functionality is delivered with infrequent crashes	Software is "professional quality"	< 2.9
4	Ability to create effective program documentation	Documentation is incomplete, unorganized and ambiguous. Language is unclear and rambling. Too many spelling and/or grammar problems.	50% of the documentation is incomplete and/or ambiguous. Language is mostly unclear and rambling. Many spelling and/or grammar problems.	20% of the documentation is incomplete and/or ambiguous. Language is mostly clear and concise. Some spelling and/or grammar problems.	Less than 5% of the documentation is incomplete and/or ambiguous. Language is clear and concise. Few spelling and/or grammar problems.	< 2.9
5	Ability to attend team meetings and contribute towards the	Only part of the group is meeting and contributing solutions	All members of the group are attending, but only some members are	All members are meeting and contributing to project in a	All members are meeting and contributing to the project	< 2.9

	solution of technical problems.		contributing solutions	meaningful way	commensurate with their abilities	
		Only part of the group is meeting and contributing solutions	All members of the group are attending, but only some members are contributing solutions	All members are meeting and contributing to project in a meaningful way	All members are meeting and contributing to the project commensurate with their abilities	< 2.9
6	Ability to listen and consider all points of view	Frequently interrupts / doesn't finish listening	Listens patiently, but ignores input	Some members will listen and modify their opinion as appropriate	All members listen and consider all points of view and derive consensus	< 2.9
7	Ability to contribute effectively to a group presentation	No balance in workload during presentation (e.g. 1 person does most of the talking). Seems like N disjoint presentations.	Very unbalanced workload during presentation. Wrong people presenting topics (e.g. another keeps jumping in to clarify things). Some coherence between presenters.	Mostly balanced workload during presentation.	Balanced workload during presentation. All presenters are part of a seamless presentation.	< 2.9
8	Ability to create software process documents while following a defined process (e.g. Waterfall, Agile, …)	Documents are incomplete, unorganized and ambiguous. Not following any development process. No version control system guiding development.	Documents are lacking in completeness, organization and correctness. Attempting to follow a development process. Version control system	Documents are mostly complete, organized and correct. Generally following the development process. Use of a version control system to track and attribute	Documents are complete, organized and correct. Clearly following a development process. A version control system was instrumental in guiding	< 2.9

			consists of naming files with version number and/or date.	changes.	development.	

e) An understanding of professional, ethical, legal, security and social issues and responsibilities

1	Understand and apply the ACM code of ethics (or similar) and principles underlying them	Not aware of any codes of ethical behavior	Aware that there are codes and other bases for ethical behavior but does not appear to follow any of them	Understands and tries to follow codes and other bases of ethical behavior	Understands and follows the code and other bases of ethical behavior	< 2.9
2	Understands and honors the property rights of others (IP, Copyright, etc)	Has been caught turning in the work of others	Lax in usage and/or referencing use of others' work	Always documents source of others' work, but may not always be proper use	Always documents source of others' work and make an effort to determine if use is appropriate	< 2.9
3	Demonstrates ethical decision making	Follows no ethical standards	Uses personal value system to support actions to the exclusion of all other ethical standards	Uses personal value system to support actions, but confuses personal ethics with professional ethics	Uses personal value system to support actions, but understands the role of professional ethical standards for corporate decisions	< 2.9

f) An ability to communicate effectively with a range of audiences

1	Ability to write a technical "white paper" (e.g. "How does PCI compliance affect the design and development of a web-based store?")	A cut-and-paste of information from the internet with no organization or focus.	Some organization, but plenty of extraneous and unimportant material. Info/page ratio is low.	Mostly organized, but could be "tighter". Information/page ratio is high.	Thoughtfully organized and clearly presented. Contains executive summary if (>5 pages). Info/page ratio is optimal.	< 2.9
2	Ability to give an effective oral presentation	Presentation is unorganized, incomplete and incorrect. Student reads slides and answers questions by re-reading a bullet point. Audience feels student does not know material.	Presentation is mostly unorganized, incomplete and incorrect. Student only reads slides and can't answer questions. Audience feels student only has cursory knowledge of material.	Presentation is somewhat lacking in organization, completeness or correctness. Student mainly reads slides and is able to answer simple questions. Audience feels student knows the material.	Presentation is organized, complete and correct. Student correctly answers difficult questions with confidence. Audience feels student has mastered the material.	< 2.9
3	Ability to create effective program documentation	Documentation is incomplete, unorganized and ambiguous. Language is unclear and rambling. A plethora of spelling/grammar problems.	50% of the documentation is incomplete and/or ambiguous. Language is mostly unclear and rambling. Many spelling/grammar problems.	20% of the documentation is incomplete and/or ambiguous. Language is mostly clear and concise. Some spelling/grammar problems.	Less than 5% of the documentation is incomplete and/or ambiguous. Language is clear and concise. Few spelling/grammar problems.	< 2.9
4	Ability to create effective software process documents	Documents are incomplete, unorganized and ambiguous. No version control	Documents are lacking in completeness, organization and correctness.	Documents are mostly complete, organized and correct. Use of a version control	Documents are complete, organized and correct. A version control system	< 2.9

		system guiding development.	Version control system consists of naming files with version number and/or date.	system to track and attribute changes.	was instrumental in guiding development.	

g) An ability to analyze the local and global impact of computing on individuals, organizations, and society

h) Recognition of the need for and the ability to engage in continuing professional development

1	Ability to work independently on complex problems	Student requires detailed or step-by-step instructions.	Student requires large amounts of guidance.	Requires small amounts of guidance	Demonstrates the ability to learn independently	< 2.9
2	Ability to research problems beyond the material covered in class	Student gives up after googling for 10 minutes.	Student just reads the Wikipedia page.	Student gathers a large amount of material (some extraneous) but does not organize or synthesize the information.	Student seeks information from multiple, independent sources and only includes relevant information.	< 2.9

i) An ability to use current techniques, skills, and tools necessary for computing practice

1	Ability to write code without bugs (e.g. Exception Safety)	Doesn't work for standard inputs	Works for standard input, doesn't handles exceptions well	Recognize all cases, may not handle some errors appropriately	Handles all cases, language security vulnerabilities	< 2.9
2	Ability to optimize the performance of a program	Can't determine performance. (e.g. "slow")	Can manually time (e.g. stopwatch - 1 or 2 seconds)	Measure in a repeatable way.	Can tune program and explain how performance was	< 2.9

					improved.	
3	Ability to effectively use a version control system to develop software	Won't use version control even if instructed to	Will use version control when instructed, but not used in an integral way	Will use version control as appropriate without being told	Can teach others in the class how to effectively use version control	< 2.9

j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

1	Ability to select the proper data structure to solve a problem	Selected a data structure that is inappropriate (likely one the student is most familiar with)	Selected a data structure that will work, but is not optimal and requires more code to implement	Selected the optimal data structure from the basic types	Designed a hybrid data structure that was superior to the basic types	
		Selected a data structure that is inappropriate (likely one the student is most familiar with)	Selected a data structure that will work, but is not optimal and requires more code to implement	Selected the optimal data structure from the basic types	Designed a hybrid data structure that was superior to the basic types	
2	Ability to recognize the complexity class of a problem	Determines the wrong complexity class	Determines the correct complexity class by guessing or w/o proper justification	Determines correct complexity class based on best solution they can derive	Determines correct complexity class and provides justification (e.g. decision trees)	
3	Ability to design an efficient algorithm to solve a problem (Use	Algorithm fails on non-standard inputs and is lacking in elegance	Algorithm works on standard inputs but is lacking in elegance or	Algorithm is optimal efficiency class, but may not be elegant or	Algorithm is optimal efficiency class, elegant and handles all input	

	same/similar for all levels, but should allow improvement as students gain mastery of CS)	and efficiency	efficiency	handle all input		
		Algorithm fails on non-standard inputs and is lacking in elegance and efficiency	Algorithm works on standard inputs but is lacking in elegance or efficiency	Algorithm is optimal efficiency class, but may not be elegant or handle all input	Algorithm is optimal efficiency class, elegant and handles all input	
		Algorithm fails on non-standard inputs and is lacking in elegance and efficiency	Algorithm works on standard inputs but is lacking in elegance or efficiency	Algorithm is optimal efficiency class, but may not be elegant or handle all input	Algorithm is optimal efficiency class, elegant and handles all input	

k) An ability to apply design and development principles in the construction of software systems of varying complexity

1	Ability to design a software system based solely on a Software Requirements Document	Most design choices are wrong (e.g. which DB to use)	Design choices mostly correct, but no documented rationale for each choice	Design choices are correct, but little rationale is documented or is an afterthought	Design choices are well thought out with rationale for each documented	
2	Ability to implement a software system	Only partial functionality is delivered	Most functionality is delivered, but program crashes too often	All important functionality is delivered with infrequent crashes	Software is "professional quality"	
3	Ability to effectively use a version control system to develop software	No version control system guiding development.	Version control system consists of naming files with version number and/or date.	Use of a version control system to track and attribute changes.	A version control system was instrumental in guiding development.	< 2.9

Appendix B – Course Syllabi

Computer Science Course Syllabi	
Course number & name:	CS 201 Computer Science I
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator:	Dr. Brian Hay
Textbook:	Starting Out With C++: From Control Structures through Objects by Tony Gaddis 2008
Specific Course Information	The discipline of computer science including problem solving, algorithm development, structured programming, top-down design, good programming style, object-oriented programming and elementary data structures. Concepts implemented with extensive programming experience in a structured language and with a group programming project.
Prerequisites or co-requisites	One year high school level programming or CS F103 and mathematics placement at the F200-level.
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes Upon completion of the course, students will be able to: • Understand and write C++ programs involving mainly C programming constructs. • Apply basic programming skills and techniques. • Apply some fundamental algorithms for searching and sorting. • Be proficient at using a development environment to write, test, and debug programs. • Describe common data structures. • Demonstrate understanding of the object-oriented programming paradigm. • Apply iteration and recursion. • Translate assigned exercises into programs using structured programming, top-down design and good programming style.
	Student Outcomes: a, b, c, i, j, k
Brief list of topics to be covered	• Basic I/O • Control Structures (selection & iteration) & Invariants • Functions (scope, pass-by-reference, overloading) • Recursion • Arrays • Pointers • Strings (C-string and standard string class)

	• Searching and Sorting 15 weekly laboratory sessions to work on assigned exercises and reinforce class lecture material. Some individual assignments, some group projects.

Computer Science Course Syllabi	
Course number & name:	CS 202 Computer Science II
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator:	Dr. Brian Hay
Textbook:	Starting Out With C++: From Control Structures through Objects by Tony Gaddis 2008
Specific Course Information	The discipline of computer science including problem solving, algorithm development, structured programming, top-down design, good programming style, object-oriented programming and elementary data structures. Concepts implemented with extensive programming experience in a structured language and with a group programming. project.
Prerequisites or co-requisites	CS 201
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes Upon completion of the course, students will be able to: • Apply problem solving skills using the object-oriented paradigm in the C++ programming language. • Extend the definition of functions and operators so that they work with different arguments. • Write abstract data types. • Hide information by limiting access to functions and variables. • Design general functions and types that can be used with different data types. • Define and use arrays, strings and vectors. • Define and manipulate linked lists and stacks. • Associate multiple meanings to one function name. • Apply good software engineering practices working on a group programming project.
	Student Outcomes: a, b, c, d, f, i, j, k
Brief list of topics to be covered:	• Classes and data abstraction (6 hours) • Operator Overloading (5 hours) • Templates (5 hours) • Inheritance (5 hours) • Polymorphism (5 hours) • Data Structures (5 hours) • Object-Oriented Design (2 hours)

	Ten individual programming projects (10 weeks) One group project (6 weeks)

Computer Science Course Syllabi	
Course number & name:	CS 301 Assembly Language Programming
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator:	Dr. Orion Lawlor
Textbook: (OPTIONAL)	Introduction to 80x86 Assembly Language and Computer Architecture by Richard Detmer 2008
Specific Course Information	Organization of computer registers, I/O and control. Digital representation of data. Symbolic coding, instructions, addressing modes, program segmentation, linkage, macros and subroutines
Prerequisites or co-requisites	CS 201
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes: To understand the relationship between the actual physical hardware and a user's code written in a high-level language (C++, Java, etc.). This allows students to write low-level code, understand and improve their high-level code's performance, debug and fix their code, and prepare their understanding of machine architecture for more advanced courses.
	Student Outcomes: a, b, c, i, j, k
Brief list of topics to be covered	• Bits, bytes, and big- and little-endian words (3 hours) • Von Neumann machine architecture and instruction encoding (3 hours) • Integer arithmetic operations & wraparound (3 hours) • Hardware and software pipelining (3 hours) • Flow control: jumps, branches, and condition codes (3 hours) • Memory: address decoding, addressing modes (3 hours) • Arrays, pointers, and address arithmetic (3 hours) • Cache sizes, speeds, and organization (2 hours) • Procedures: call/return, the stack, parameter passing (3 hours) • Mixing Assembly, C/C++, and Fortran (2 hours) • Multiple-precision arithmetic (1 hour) • Floating-point representation (3 hours) • SIMD extensions: SSE, AltiVec (2 hours) Laboratory projects (specify number of weeks on each)

	• Read big-endian binary input file on little-endian machine (one week) • Design a simple instruction encoding and build a CPU simulator for that encoding (two weeks) • Relate C and assembly code; measure performance improvement from software pipelining on real machines (two weeks) • Write assembly to access memory; measure performance improvement from cache locality (two weeks) • Use assembly to fetch the frame pointer, and write C/C++ code to unwind the execution stack (two weeks) • Measure performance improvement from SIMD extensions (two weeks)

Computer Science Course Syllabi	
Course number & name:	CS 311 Data Structures and Algorithms
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator:	Dr. Chris Hartman
Textbook:	None – slides and reading materials provided
Specific Course Information	Data structures and the algorithms for their manipulation. Object-oriented programming, arrays, tables, stacks, queues, trees, linked lists, sorting, searching and hashing.
Prerequisites or co-requisites	CS 202
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes: After taking the course, students should • Have improved proficiency in programming and software design, particularly advanced C++ (exceptions and exception safety, STL, const correctness). • Understand what an abstract data type is, and how ADTs relate to software design. • Be able to apply the basic ideas behind analysis of algorithmic efficiency, including use of "big-*O*" notation. • Be familiar with various standard algorithms, including those for sorting and searching. • Be familiar with standard data structures, including knowledge of their implementations and relevant trade-offs.
	Student Outcomes: a,c, i, j, k
Brief list of topics to be covered	• Advanced C++ • Software Design Principles • Recursion • Abstract Data Types • Algorithmic Efficiency • Sorting • Smart Arrays & Strings • Linked Lists • Stacks & Queues • Trees • Priority Queues

	• Tables Weekly programming assignments.

Computer Science Course Syllabi	
Course number & name:	CS 321 Operating Systems
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Glenn Chappell
Textbook:	None
Specific Course Information	Functions of files and operating systems. Review of required architectural features. The PROCESS concept. Storage management, access methods and control, interrupt processing, scheduling algorithms, file organization and management, and resource accounting.
Prerequisites or co-requisites	CS F301
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes: • Know the purposes and responsibilities of operating systems. • Be familiar with various standard operating systems concepts (interrupt, system call, process, thread, scheduling, concurrency, deadlock, virtual memory, etc.). • Understand important ways in which major operating systems differ. • Be familiar with operating system implementation issues involved in process management, memory management, and file systems. • Have a basic programming proficiency in the use of system calls, multiple threads, and memory mapping.
	Student Outcomes: a,b,c,d,f,i
Brief list of topics to be covered	• processes (scheduling, threads, concurrency) • memory (allocation, virtual memory) • file systems • networking

Computer Science Course Syllabi	
Course number & name:	CS 331 Programming Languages
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Glenn Chappell
Textbook:	None
Specific Course Information	Syntax and semantics of widely differing programming languages. Syntax specification, block structure, binding, data structures, operators and control structures. Comparison of several languages such as ALGOL, LISP, SNOBOL and APL.
Prerequisites or co-requisites	CS F311
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes: • Understand the concepts of syntax and semantics, and how these can be specified. • Understand basic lexical analysis and parsing, and have experience writing code to do these. • Understand the various kinds of languages and the primary ways in which they differ. • Understand various standard language features and forms they can take in different languages. • Have a basic programming proficiency in multiple significantly different languages.
	Student Outcomes: a, d, f, h, i, j
Brief list of topics to be covered	• Specification • Parsing • lexical analysis • classes of languages • various specific languages and language features (types, identifiers, etc.)

Computer Science Course Syllabi	
Course number & name:	CS 361 Systems Security and Administration
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator	Dr. Brian Hay
Textbook:	Essential System Administration (3rd Edition) by Frisch Practical Unix & Internet Security (3rd Edition) by Garfinkel, Spafford, and Schwartz. This course uses Blackboard and the ASSERT Lab.
Specific Course Information	Advanced systems programming including privileged instructions and system services, authentication technologies, host-based and network-based security issues. Applications to asynchronous I/O, process control and communication, device drivers and file management.
Prerequisites or co-requisites	CS 301
Required, elective or select elective	Elective

Specific goals for the course:	Instruction Outcomes: Upon completion of the course, students will be able to: • Understand common system administration topics, and the security challenges/implications of those topics. • Understand generic system administration topics (e.g., User Authentication) at a theoretical level and in the context of several specific systems. • Understand common security issues, including their root causes and common methods to address them.
Student Outcomes	(a), (b), (c), (d), (i), (k)
Brief list of topics to be covered	• Overview of System Administration and Security • Operating System Installation, Configuration, and Management • Software Installation and Update Management • User Management • Filesystem Management • Logging and System Monitoring • Networking • Networked Applications/Services/Servers • Source Code Review Weekly laboratory sessions to work on assigned exercises and reinforce lecture material. Some individual assignments and a group project.

Computer Science Course Syllabi	
Course number & name:	CS 381 Computer Graphics
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Glenn Chappell
Textbook:	None - Supplementary readings will be provided by instructor.
Specific Course Information	Creation of computer-generated images on programmable 3-D graphics hardware. Color, lighting, textures, hidden surfaces, 3-D geometric transformations, curve and surface representations, 2-D and 3-D user interfaces, and the visual modeling of physical phenomena.
Prerequisites or co-requisites	CS F202; MATH F202X or MATH F314.
Required, elective or select elective	Elective
Specific goals for the course:	Instruction Outcomes: • Have a good overall understanding of 2-D/3-D graphics programming based on the synthetic-camera model, using rendering-pipeline hardware. • Be familiar with a professional-quality 2-D/3-D graphics API. • Know how to program modern graphics hardware using a high-level shading language. • Understand how to do simple event-driven programming. • Understand various issues involved in 3-D graphics: transformations, hidden-surface removal, lighting. • Understand and be able to use facilities for rendering complex scenes using simple primitives.
	Student Outcomes: a,c, f, h, i, j
Brief list of topics to be covered	• rendering pipeline • animation • event-driven interaction • mouse-based interfaces • transformations • 3-D viewing & hidden-surface removal • shaders & GLSL • lighting, texturing & related effects • simple simulations (“physics”) representing curves and surfaces

Computer Science Course Syllabi	
Course number & name:	CS 405 Introduction to Artificial Intelligence
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Jon Genetti
Textbook: Required	Artificial Intelligence, by Lugar 2005
Specific Course Information	Examine diverse branches of Al placing Al in larger context of computer science and software engineering. Knowledge representation formalism and search technology. Programming methodologies; procedural systems such as expert systems and blackboard systems and non-procedural systems such as neural networks. Software engineering aspects of problem selection, knowledge acquisition, verification and validation. Individual projects.
Prerequisites or co-requisites	CS F311 or permission of instructor.
Required, elective or select elective	Elective

Specific goals for the course:	✓ The student will understand how to formulate an AI problem and search for an optimal solution. ✓ The student will understand when/how to apply heuristics to problems and/or searches. ✓ The student will implement a project using neural networks and evolutionary programming.
Student Outcomes:	✓ (A) An ability to apply knowledge of computing and mathematics appropriate to the discipline ✓ (B) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution ✓ (C) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired goals ✓ (I) An ability to use current techniques, skills, and tools necessary for computing practice ✓ (J) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

Brief list of topics to be covered:	• Problem Formulation • Brute-force Searching • Heuristic Searching • Game Searching • Evolutionary Programming • Neural Networks • Logic Programming

Computer Science Course Syllabi	
Course number & name:	CS 411 Analysis of Algorithms
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Chris Hartman
Textbook:	The Design and Analysis of Algorithms, Anany Levitin 2006
Specific Course Information	Analysis of classic algorithms, their implementation and efficiency. Topics from combinatorics (sets, graphs), algebra (integer arithmetic, primes, polynomial arithmetic, GCD, Diophantine equations, encryption), systems (parsing searching, sorting) and theory (recursion, Turing machines). The complexity classes P, NP and NP complete.
Prerequisites or co-requisites	CS 311 and Math 307
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes: • Be able to apply design and analysis techniques to algorithms • Be able to determine when algorithms for a particular problem are optimal • Be able to determine the algorithmic complexity for a new problem
	Student Outcomes: a,c,j,k
Brief list of topics to be covered	• Recurrence Relations • Brute-force Algorithms • Divide-and-Conquer Algorithms • Decrease-and-Conquer Algorithms • Transform-and-Conquer Algorithms • Space and Time Tradeoffs • Dynamic Programming • Greedy Technique • Limitations of Algorithm Power • NP and NP Complete No programming projects. Weekly homework is assigned.

Computer Science Course Syllabi	
Course number & name:	CS 425 Database Systems
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator	Dr. Brian Hay
Textbook:	Database Design. Application Development, and Administration by Mannino (Third Edition) This course uses Blackboard and the ASSERT Lab.
Specific Course Information	Data independence, modeling, relationships and organization. Hierarchical, network and relational data models; canonical schema. Data description languages, SQL, query facilities, functional dependencies, normalization, data integrity and reliability. Review of current database software packages.
Prerequisites or co-requisites	CS311, CS 321
Required, elective or select elective	Elective

Specific goals for the course:	Instruction Outcomes: Upon completion of the course, students will be able to: • Understand the relational data model • Develop database designs • Develop database driven/dependent applications • Understand basic database administration • Understand common database security issues
Course Outcomes	(a), (b), (c), (d), (i), (k)
Brief list of topics to be covered	• Introduction to Database Environments • Understanding Relational Databases • Data Modeling • Relational Database Design • Application Development with Relational Databases • Database Administration • Database Security Weekly laboratory sessions to work on assigned exercises and reinforce lecture material. Some individual assignments and a group project.

Computer Science Course Syllabi	
Course number & name:	CS 431 Programming Language Implementation
Credits and contact hours:	3+0 credits 39.9 hours
Instructor:	Dr. Jon Genetti
Textbook: Required Optional	Modern Complier Implementation in C, by Andrew W. Appel 2004 Lex and Yacc, by Doug Brown, John Levine and Tony Mason 1992
Specific Course Information	Design and implementation of major phases of high level language translators including scanning, parsing, translation, code generation and optimization. Students develop a compiler for a language in a group project which emphasizes good software engineering practices in structured design, testing and documentation.
Prerequisites or co-requisites	CS F331; ENGL F111X; ENGL F211X or ENGL F213X
Required, elective or select elective	Elective

Specific goals for the course:	✓ The student will understand the compilation process, which includes lexical analysis, syntax-directed translation, syntax analysis and code generation. ✓ The student will be able to use common compiler generation tools such as lex/flex and yacc/bison. ✓ The student will be able to create a working compiler from a language specification.
Student Outcomes:	✓ (C) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs ✓ (I) An ability to use current techniques, skills, and tools necessary for computing practice ✓ (K) An ability to apply design and development principles in the construction of software systems of varying complexity
Brief list of topics to be covered:	• Structure of a Compiler • Lexical Analysis • Syntax Analysis • Semantic Analysis • Intermediate Code Generation • Code Optimization • Code Generation • Symbol-Table Management • Compiler-Construction Tools

Computer Science Course Syllabi	
Course number & name:	CS 441 Computer Architecture
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Orion Lawlor
Textbook: (Optional)	Computer Organization and Design: The Hardware/Software Interface by John Hennessy, Morgan Kaufmann and David Patterson 2004
Specific Course Information	Computer design fundamentals, performance and cost, pipelining, instruction-level parallelism, memory hierarchy design, storage systems, and vector processing.
Prerequisites or co-requisites	CS F321; EE F341.
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes: KNOWLEDGE: To understand the operation of the major components of generic computer systems, how they work together, and the impact of evolving technology on these components (e.g., memory, processor, and I/O), To understand a number of commonly used computer architecture metrics, to understand salient features of the design of multiprocessors and highly parallel machines, and To understand selected current events and "buzzwords" in the area of computer systems architecture (e.g., "fault tolerant" and "multimedia" computers, 64-bit computers) SKILLS: To be able to conduct design and analysis exercises to optimize various computer architecture measures, and to evaluate tradeoffs among these measures.
	Student Outcomes: a, b, c, h, i, j
Brief list of topics to be covered	• Computer abstractions and technology (new technology -64 bit) • Performance (benchmarks, other measures (5 hours) • Instruction sets (3 hours) • Computer arithmetic (3 hours) • Datapath and control (6 hours) • Pipelining (5 hours) • Memory hierarchies (6 hours) • Multiprocessors (3 hours) • Student project presentations (6 hours) Student complete individual projects that are self selected (6 weeks)

Computer Science Course Syllabi	
Course number & name:	CS 442 Computer Communication and Networks
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator	Dr. Brian Hay
Textbook:	Computer Networking: A Top Down Approach by Kurose and Ross 2007 This course uses Blackboard and the ASSERT Lab.
Specific Course Information	Study of computer networks using the ISO/OSI layered models as a framework. Design issues and trade-offs, protocols and selected standards. Emphasis on ISO/OSI Layers 1-4/(Physical, Data Link, Network and Transport Layers), plus medium access sublayers (LAN's, etc.).
Prerequisites or co-requisites	CS 321
Required, elective or select elective	Elective
Specific goals for the course:	Instruction Outcomes: Upon completion of the course, students will be able to: • Understand modern computer communication networks, including the concepts of encapsulation and a layered network stack. • Understand and implement networking at the application layer. • Understand transport layer protocols, including TCP and UDP. • Understand Network/Internet Layer protocols, including IPv4 and IPv6. • Understand and configure routing protocols. • Understand link layer protocols. • Understand security issues at each of the layers in the network stack.
Student Outcomes	(a), (b), (c), (d), (i), (k)
Brief list of topics to be covered	• Overview of Computer Networking • Application Layer • Transport Layer • Network/Internet Layer • Wireless and Mobile Networking • Network Security • Network Management Weekly laboratory sessions to work on assigned exercises and reinforce lecture material. Some individual assignments and a group project.

Computer Science Course Syllabi	
Course number & name:	CS 451 Automata and Formal Languages
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Glenn Chappell
Textbook:	Hopcroft, Motwani, and Ullman, Automata Theory, Languages, and Computation, 3rd ed., Addison-Wesley, 2007
Specific Course Information	Finite automata, regular languages, phrase structured grammars, context free language, push down automata, deterministic context free languages, recursive and recursively enumerable languages, Turing machines, decision problems, and undecidability.
Prerequisites or co-requisites	MATH F307; CS F201
Required, elective or select elective	Required
Specific goals for the course:	Instruction Outcomes: • Understand the basic models of computation: finite automata, push-down automata, and Turing machines. • Understand the connection between automata and the languages they accept and the computational power required to solve particular problems. • Have experience in proving decidability results. • Recognize the well known intractable problems and decidability results.
	Student Outcomes: a, b, j
Brief list of topics to be covered	• introduction to proof • automata • finite automata • regular expressions • regular languages • push-down automata • context-free grammars • context-free languages • Turing machines and decidability • intractable problems

Computer Science Course Syllabi	
Course number & name:	CS 460 Introduction to Digital Forensics
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator	Dr. Brian Hay
Textbook:	Real Digital Forensics: Computer Security and Incident Response, by Keith J. Jones, Richard Bejtlich and Curtis W. Rose. 2005 This course uses Blackboard and the ASSERT Lab.
Specific Course Information	Takes a hands-on approach to the forensics examination of computer technology. Focuses on the forensic process, methods, and tools utilized to collect and preserve and examine digital evidence. Course topics include: collection, preservation and examination of evidence from computers including file systems, email and malicious code.
Prerequisites or co-requisites	CS 321 or permission of instructor.
Required, elective or select elective	Elective

Specific goals for the course:	Instruction Outcomes: Upon completion of the course, students will be able to: • Identify digital devices and systems/networks that are likely to include data of relevance to an investigation. • Acquire data from digital devices and systems/networks in a manner appropriate to an investigation. • Analyze recovered data in order to produce evidence in a forensic investigation. • Understand the basic operation and limitations of existing commonly used forensics tools for the acquisition, analysis, and presentation of digital data/evidence. • Produce custom tools to address specific challenges that are not addressed by existing digital forensics tools. • Understand and discuss some of the challenges currently facing the digital forensics community. • Discuss the basic legal and ethical issues related to digital forensics.
Student Outcomes	(a), (b), (c), (d), (e), (i), (k)

Brief list of topics to be covered	• Overview of Digital Forensics • Live Incident Response • Network-Based Forensics • Forensic Duplication • General File System Forensics • Web and Email Forensics • Windows Registry Examination • Analysis of Unknown Files and Malware • Mobile Device Forensics Weekly laboratory sessions to work on assigned exercises and reinforce lecture material. Some individual assignments and a group project.

Computer Science Course Syllabi	
Course number & name:	CS 462 Intrusion Detection Systems
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator	Dr. Brian Hay
Textbook:	Intrusion Detection by Bace 2000. This course uses Blackboard and the ASSERT Lab.
Specific Course Information	Focus on IDS theory and practice and its importance; the origin and resolution of common security threats and vulnerabilities; host and network approaches to IDS implementation; and the legal, ethical, and privacy issues associated with IDS use and policies.
Prerequisites or co-requisites	CS 361 or permission of instructor.
Required, elective or select elective	Elective

Specific goals for the course:	Instruction Outcomes: Upon completion of the course, students will be able to: • Understand the major classifications of intrusion detection systems, including their strengths and weaknesses • Understand the threats that IDS are and are not effective in combating. • Integrate IDS into a security architecture to meet a set of security goals. • Understand the role of honeynets and honeypots in a security architecture, including their use in conjunction with IDS. • Select appropriate sources of data for IDS in a given environment. • Understand and select IDS response mechanisms for a given environment. • Perform analysis of IDS results/alerts, and investigate the underlying issue that triggered the alert. • Discuss common legal and ethical issues related to the use of IDS.
Student Outcomes	(a), (b), (c), (d), (e), (i), (k)

Brief list of topics to be covered	• Overview of IDS • Information Sources • Analysis Schemes • Response Strategies • Honeypots and Honeynets • Virtualization and Virtual Machine Introspection • Current Trends and Challenges • Legal and Ethical Issues Weekly laboratory sessions to work on assigned exercises and reinforce lecture material. Some individual assignments and a group project.

Computer Science Course Syllabi	
Course number & name:	CS 463 Cryptography and Data Security
Credits and contact hours:	3+0 credits 45 hours
Course Coordinator	Dr. Brian Hay
Textbook:	Applied Cryptography by Bruce Schneier (Second Edition). This course uses Blackboard and the ASSERT Lab.
Specific Course Information	Specialized study of cryptography and its application in securing data systems, with an emphasis on applied cryptography. Topics include history of cryptography, encryption, digital signatures, authentication, electronic commerce, key distribution and management, private and public key cryptography, and protocols.
Prerequisites or co-requisites	MATH F307; CS F311; or permission of instructor.
Required, elective or select elective	Elective

Specific goals for the course: (specific outcomes of instruction. ex: the student will be able to explain the significance of current research about a particular topic.)	Instruction Outcomes: Upon completion of the course, students will be able to: • Understand and be able to discuss the weaknesses of common historical cryptographic algorithms. • Understand the major types of modern cryptography. • Select appropriate modern cryptographic algorithms to address the requirements of a particular environment/system. • Understand the basic approaches to cryptanalysis of historical and modern cryptographic algorithms. • Use appropriate cryptographic algorithms in the development of new systems/software. • Understand and discuss common pitfalls in the implementation of cryptographic algorithms.
Student Outcomes	(a), (b), (c), (d), (i), (k)
Brief list of topics to be covered	• Basic Concepts • Historical Cryptography • Block Ciphers • Stream Ciphers • Hash Functions • Asymmetric Key Cryptography • Digital Signatures

Brief list of topics to be covered continued	• Key Exchange • Cryptanalysis Weekly laboratory sessions to work on assigned exercises and reinforce lecture material. Some individual assignments and a group project.

Computer Science Course Syllabi	
Course number & name:	CS 471 Software Engineering
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Jon Genetti
Textbook:	Software Engineering by Ian Sommerville 2008
Specific Course Information	Introduction to basic software engineering principles, techniques, methods and standards as applied to the engineering of complex software systems. Topics from software system development process models, multiple view system modeling and specification using UML, classification of software systems, project management and legal issues.
Prerequisites or co-requisites	CS F311; ENGL F111X; ENGL F211X or ENGL F213X
Required, elective or select elective	Required

Specific goals for the course:	✓ The student will be able to create a Software Requirements Document from customer specifications. ✓ The student will be able to design and implement a software project using a version control system while creating the appropriate documents. ✓ The student will be able to write a technical "white paper." ✓ The student will be able to give an effective oral presentation. ✓ The student will be able to create effective program documentation.
Student Outcomes:	✓ (B.3/K.1) Ability to create a software requirements document ✓ (C.4) Ability to design a software system based solely on a Software Requirements Document ✓ (C.5) Ability to implement a software system ✓ (F.1) Ability to write a technical paper ✓ (F.2) Ability to give an effective oral presentation ✓ (F.3) Ability to create effective program documentation ✓ (F.4) Ability to create effective software process documents ✓ (K.3) Ability to effectively use a version control system to develop software

Brief list of topics to be covered:	• Software Processes • Project Management • Software Requirements • Requirements Engineering Processes • System Models • Architectural Design • Distributed Systems Architectures • Application Architectures • Real-time Software Design • User Interface Design • Agile Software Development • Verification and Validation • Software Testing

Computer Science Course Syllabi	
Course number & name:	CS 472 Senior Project and Professional Practice
Credits and contact hours:	3+0 credits 39.9 hours
Instructor:	Dr. Jon Genetti
Textbook: Required	Software Engineering, by Ian Sommerville 2008
Specific Course Information	Group projects in a real computer industry environment and produce appropriate documentation and reports. Nature, ethics, and legal considerations of the computer science profession are discussed with an emphasis on ethics. Additional topics include project management, design methodologies, technical presentation, human-machine interface and programming team interactions.
Prerequisites or co-requisites	Senior Standing; CS F471; COMM F131X or COMM F141X; ENGL F111X; ENGL F211X or ENGL F213X or permission of instructor
Required, elective or select elective	Required

Specific goals for the course:	✓ The student will contribute to a team effort to analyze, implement and deliver a software system to a real-world client. ✓ The student teams will effectively use a version control system and create the appropriate documentation for the software system. ✓ The student will demonstrate ethical decision making and understand IP issues in the computer field.
Student Outcomes:	✓ (D.1) Ability to create a Software Requirements Document for a real-world client ✓ (D.2) Ability to design a large software system for a real-world client ✓ (D.3) Ability to implement and deliver a large software system for a real-world client ✓ (D.4) Ability to create effective program documentation ✓ (D.5) Ability to attend meetings and contribute towards the solution of technical problems ✓ (D.6) Ability to listen and consider all points of view ✓ (D.7) Ability to contribute effectively to a group presentation ✓ (D.8) Ability to create software process documents while following a defined process (e.g. Waterfall, Agile, …) ✓ (E.1) Understand and apply the ACM Code of Ethics and principles underlying them ✓ (E.2) Understand and honor the property rights of others

	✓ (E.3) Demonstrate ethical decision making
Brief list of topics to be covered:	• Review of Proposed Projects • Project Selection and Team Formation • Review of Software Processes (Waterfall, Agile, etc) • Project Management • Version Control Systems • Ethics in Computer Science • Intellectual Property and Software Patents • Non-Disclosure and Non-Compete Agreements

Computer Science Course Syllabi	
Course number & name:	CS F481 Topics in Computer Graphics
Credits and contact hours:	3+0 credits 45 hours
Instructor:	Dr. Glenn Chappell
Textbook:	None
Specific Course Information	Hardware, software and techniques used in computer graphics taken from topics such as volume rendering, particle systems, shading, image processing, computer aided design, video effects, animation and virtual environments.
Prerequisites or co-requisites	CS F381
Required, elective or select elective	Elective
Specific goals for the course:	Instruction Outcomes: • Use C++/OpenGL and programmable graphics hardware to write modern graphics software. • Understand current technqiues in the rendering field.
	Student Outcomes: a, c, h, i, j
Brief list of topics to be covered	Writing C++ OpenGL applications with programmable GLSL shaders that run on the graphics card using raytracing, volume rendering, soft shadows, antialiasing, and radiosity

Computer Science Course Syllabi	
Course number & name:	CS F490 Student Internship
Credits and contact hours:	1 - 3 credits
Instructor:	Dr. Peter Knoke
Textbook:	None
Specific Course Information	Students work on computer science project under the joint direction of a faculty member and participating industry or governmental agency.
Prerequisites or co-requisites	Junior standing and acceptance in an approved internship program.
Required, elective or select elective	Elective
Specific goals for the course:	Instruction Outcomes: Course Goals Project goals are specific relevance to both core computer science technical areas and significant technical needs of the participating industry or government agency. Instruction Outcomes Both the directing faculty member and a representative of the participating industry/agency agree that the student efforts have been helpful to the industry/agency and demonstrate good research and understanding of the project's key technical areas.
(Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course)	Student Outcomes: a, b, c, d, e, f, g, h, i, j, k are all possible, depending on the project specifics.
Brief list of topics to be covered:	Major Topics Covered in the Course Topics vary depending on the specific project and its goals. For example, project areas could include software application areas of interest to local companies [web sites], or software application areas important for federal agencies like the Bureau of Land Management [cabin management], or Denali National Park [annual road lottery], or the National Security Agency [securitymethods].

Course number &	CS 482 Simulations in Computer Graphics (2009: CS 480; 2011: CS 493)
Credits and contact hours:	3+0 credits 39.9 hours
Instructor:	Dr. Orion Lawlor
Textbook:	None. Detailed derivations and example code provided on course website.
Specific Course Information	Software to simulate physical phenomena for use in interactive visualization, such as particle systems, Navier-Stokes fluid dynamics, and finite element solid mechanics. Includes Lagrangian and Eulerian meshes, stability, and discretization order. For interactive graphics use, high performance qualitatively correct simulations are more useful than high-precision solutions. topics such as volume rendering, particle systems, shading, image processing, computer aided design, video effects, animation and virtual environments.
Prerequisites or co-requisites	CS F381; Physics 212X
Required, elective or select elective	Elective
Specific goals for the course:	By the end of the course, you will be able to build and understand simulators for a variety of physical phenomena, including moving fluids and solids. Along the way, you will learn how to use both moving Lagrangian and non-moving Eulerian meshes, how to discretize partial differential equations in space and time, how to keep your simulations computationally stable (prevent them from "blowing up"!) and how to apply that knowledge in a variety of domains. Basic graphics programming in C++ and GLSL, basic Newtonian physics, and good familiarity with calculus are all required.
Student Outcomes	• (d): Students will work on a group project during this course. • (f): Students will orally present their semester projects. • (h): Students will learn new GPU techniques, and learn how to learn new techniques.

Brief list of topics to be covered	• Particles (2 weeks)
	  • Basic OpenGL review. High-performance rendering, framebuffer and vertex buffer objects, programmable shading via GLSL, point sprites [HW1]
	  • Particle systems: from Newton's Laws to computer code. Discretizing partial differential equations, time integration, discretization error, stability
	  • Forces: gravity, friction, user interface, flocking [HW2]
	  • Boundary conditions: bounding particles with planes, spheres, cylinders. Boundary enforcement.
	• 2D Grids (4 weeks)
	  • OpenGL 2D texturing review: nearest-neighbor, bilinear, mipmapping, anisotropic mipmapping/filtering
	  • Shallow-Water Wave Equation [HW3]
	  • Turk/Turing Reaction-Diffusion Equations
	  • Continuous to discrete transformations on the plane: discretizing the gradient operator [HW4]
	  • Courant stability limit, speed of sound
	  • Boundary condition images / geometric boundary condition enforcement [HW5]
	  • Simulating unbounded problem domains using a perspective grid
	
	After Spring Break:
	
	• 3D Grids (3 weeks)
	  • OpenGL 3D texturing review: solid texturing, framebuffer 2D slice rendering
	  • 2D/3D fluid simulation: Navier-Stokes, Stam's Stable Fluids [HW6]
	  • 3D light transport simulation
	  • Speeding up slow codes: smaller storage types, optimizing GPU arithmetic, multigrid [HW7]
	• FEM & unstructured geometry (4 weeks)
	  • OpenGL vertex & element buffer objects
	  • Stress/strain, elastic & plastic behavior
	  • Element orders & shape functions
	  • 2D, plate, and volume elements [HW8]
	  • Unstructured boundary conditions
	  • Failure & fracture simulations [HW9]

Appendix C – Equipment

Most classrooms typically include a laptop-associated projector, an overhead projector and a blackboard or whiteboard. Where a laptop-associated projector is provided, faculty normally use a personal laptop. Other equipment is described under Criteria 7A and 7B.

Appendix D – Institutional Summary

1. The Institution

a. Name and address of the institution

University of Alaska Fairbanks
PO Box 757500
Fairbanks, Alaska 99775-7500
www.uaf.edu

b. Name and title of the chief executive officer of the institution

Pat Gamble, UA System President
Brian Rogers, UAF Chancellor

c. Name and title of the person submitting the self-study report.

Dr. Kara Nance
Professor and Chair,
Computer Science Department

d. Name the organizations by which the institution is now accredited and the dates of the initial and most recent accreditation evaluations.

The university has been accredited by the Northwest Commission on Colleges and Universities since 1934. The most recent full-scale accreditation evaluation was in 2001. This was followed in 2006 by a five-year interim report. The next NWCCU accreditation self-study will be submitted in fall 2011.

2. Type of Control

Description of the type of managerial control of the institution, e.g., private-non-profit, private-other, denominational, state, federal, public-other, etc.

State and Federal.

3. Educational Unit

Describe the educational unit in which the program is located including the administrative chain of responsibility from the individual responsible for the program to the chief executive officer of the institution. Include names and titles. An organization chart may be included.

The College of Engineering and Mines (CEM) is organized into six departments:

a. Civil and Environmental Engineering
b. Computer Science
c. Electrical and Computer Engineering

d. Mechanical Engineering
e. Mining and Geological Engineering
f. Petroleum Engineering, and

offers the following programs

- Arctic Engineering M.S.
- Civil Engineering B.S., M.C.E., M.S.
- Computer Engineering B.S.
- Computer Science B.S., M.S.
- Construction Management graduate certificate
- Electrical Engineering B.S., M.E.E., M.S.
- Engineering Ph.D.
- Engineering Management M.S.
- Environmental Quality Engineering M.S.
- Environmental Quality Science M.S.
- Geological Engineering B.S., M.S.
- Mechanical Engineering B.S., B.S/M.S., M.S.
- Mineral Preparation Engineering M.S.
- Mining Engineering B.S., M.S.
- Petroleum Engineering, B.S., M.S.
- Science Management M.S.
- Software Engineering M.S.E.

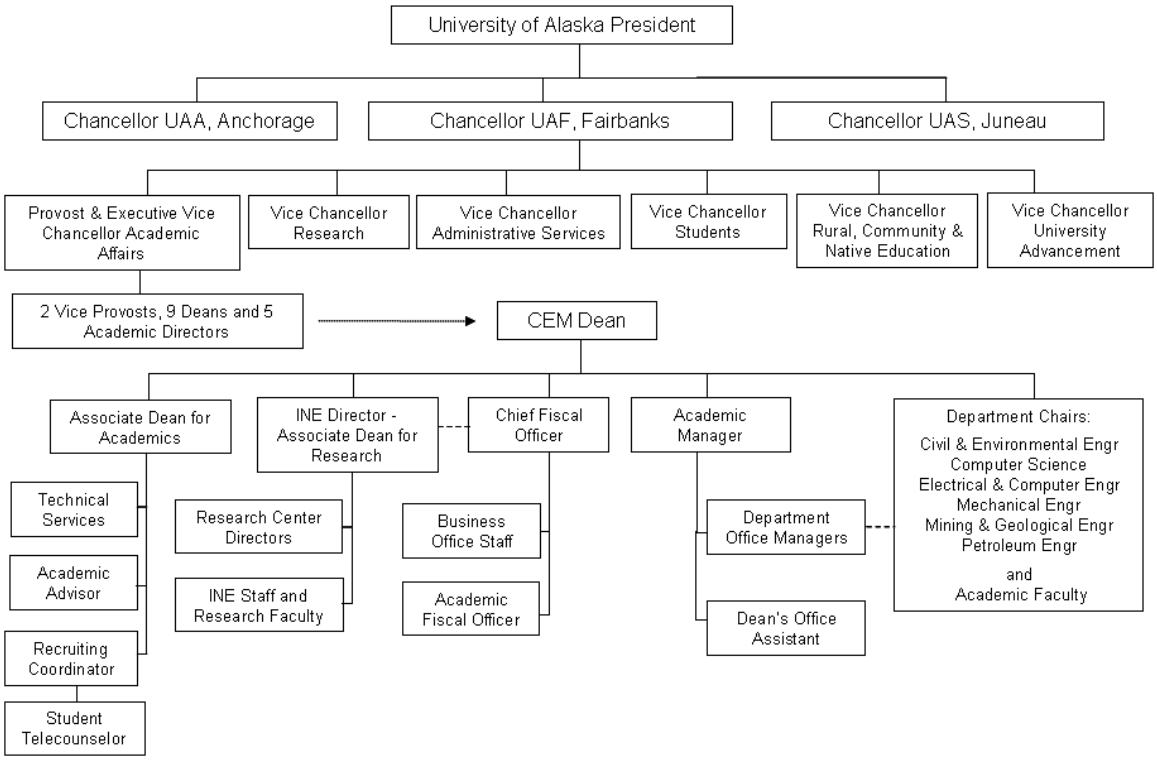
The FY 10 enrollment in the college was 672 undergraduate students and 146 graduate students, and there were 101 degrees awarded. Grant-funded research expenditures in INE (Institute of Northern Engineering) totaled $14,306,000 in FY 10, with total research expenditures of $18,184,000.

The Computer Science department joined CEM in FY11, and with the addition of their 7 faculty, CEM/INE currently has 59 faculty, including 6 that are research only, and 46.5 staff members.

The college organization chart is below. The top level administration from the chart is:
CEM Dean – Douglas Goering
Associate Dean for Instruction – Charlie Mayer
INE Director – Associate Dean for Research – Daniel White
Chief Fiscal Officer – Nickole Conley
Academic Manager – Linda Ilgenfritz
Civil and Environmental Engineering Department Chair – David Barnes
Computer Science Department Chair – Kara Nance
Electrical and Computer Engineering Department Chair – Charlie Mayer
Mechanical Engineering Department Chair – Jonah Lee
Mining and Geological Engineering Department Chair – Rajive Ganguli
Petroleum Engineering Department Chair – Catherine Hanks

The CEM Dean reports to the UAF Provost, Susan Henrichs, who reports to the UAF Chancellor, Brian Rogers, who reports to the UA President, Pat Gambel.

College of Engineering and Mines within the University of Alaska
Organizational Chart



4. Academic Support Units

List the names and titles of the individuals responsible for each of the units that teach courses required by the program being evaluated, e.g., mathematics, physics, etc.

Chemistry, John Keller, Chemistry and Biochemistry Department Chair
Math, Anthony Rickard, Mathematics and Statistics Department Chair
Physics, Ataur Chowdhury, Physics Department Chair
English, Rich Carr, English Department Chair
College of Liberal Arts (CLA), Burns Cooper, Interim Dean College of Liberal Arts. CLA offers the general education requirement courses.

5. Non-academic Support Units

List the names and titles of the individuals responsible for each of the units that provide non-academic support to the program being evaluated, e.g., library, computing facilities, placement, tutoring, etc.

Libraries [http://www.uaf.edu/uaf/academics/libraries], Responsible Individual: Paul McCarthy, Interim Dean of Libraries.

UAF computing facilities are operated by OIT (Office of Information Technology); [www.uaf.edu/oit], Responsible Individual: Steve Smith, Chief Information Technology Officer.

UAF Academic Advising Center provides placement testing and advising; [www.uaf.edu/advising], Responsible Individual: Linda Hapsmith, Director.

Tutoring is provided within the academic units. The Department of Mathematical Sciences (DMS) provides daily tutoring in the Math Lab, located in the Chapman Building [http://www.dms.uaf.edu/dms/MathLab/MathLabIntro.html], Responsible Individual: Latrice Laughlin, Instructor, Department of Mathematics and Statistics. The Engineering Tutoring Lab is located in the Duckering Building and is manned 6 days a week by engineering student tutors, Responsible Individual: Charlie Mayer, Associate Dean for Instruction, CEM.

6. Credit Unit

It is assumed that one semester or quarter credit normally represents one class hour or three laboratory hours per week. One academic year normally represents at least 28 weeks of classes, exclusive of final examinations. If other standards are used for this program, the differences should be indicated.

The standard definition of credit hour applies at UAF: 1 credit hour represents 1 hour of class per week (or three laboratory hours per week) for 14 weeks per semester.

7. Tables

Complete the following tables for the program undergoing evaluation.

Table D-1. Program Enrollment and Degree Data

Computer Science

	Academic Year		Enrollment Year					Total Undergrad	Total Grad	Degrees Awarded			
			1st	2nd	3rd	4th	5th			Associates	Bachelors	Masters	Doctorates
Current Year	AY10-11	FT	19	11	14	18		62	11	0	5	1	0
		PT	5	1	7	13		26	3				
1	AY09-10	FT	20	15	14	19		68	7	0	13	5	0
		PT	5	2	5	13		25	2				
2	AY08-09	FT	21	14	11	18		64	4	0	10	4	0
		PT	3	2	5	11		21	4				
3	AY07-08	FT	24	12	23	22		81	3	0	13	2	0
		PT	4	2	5	14		25	3				
4	AY06-07	FT	20	13	22	24		79	8	0	14	6	0
		PT	2	5	5	5		17	8				

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

FT--full time
PT--part time

UAF Institutional Research tracks enrollment past the 4th year as being in the senior year; hence there is no separate data for 5th and succeeding years.

University of Alaska Fairbanks

Degrees awarded 2011 – University Wide

Academic Year		Enrollment Year					Total Undergrad	Total Grad	Degrees Awarded			
		1st	2nd	3rd	4th	5th			Associates	Bachelors	Masters	Doctorates
2010-2011	FT	1389	750	619	851	No data collected	3609	659				
	PT	792	428	318	621		2159	500				
2009-2010	FT	1374	728	620	786		3508	640	204	473	219	45
	PT	681	386	293	535		1895	494				
2008-2009	FT	1260	688	584	746		3278	575	193	483	169	37
	PT	578	355	240	507		1680	465				
2007-2008	FT	1261	620	556	748		3185	601	238	444	199	29
	PT	516	356	259	484		1615	422				
2006-2007	FT	1237	652	592	745		3226	605	210	464	199	33
	PT	527	334	279	455		1595	457				

Table D-2. Personnel

Computer Science

Year: Fall 2010

	HEAD COUNT		FTE[2]
	FT	PT	
Administrative[3]	2		2
Faculty (tenure-track)	7		7
Other Faculty (excluding student Assistants)		1	.5
Student Teaching Assistants	4		4
Student Research Assistants	4		4
Technicians/Specialists		1	.5
Office/Clerical Employees		2	1
Others[4]		3	

Others include undergraduate students who work configuring, maintaining, and managing labs. Data above is specific to Computer Science. In addition, we have access to CEM personnel as described in section 8.C.

Appendix E – Faculty Vitae

Jon D. Genetti

Education

Degree	Field	Institution	Date
PhD	CS	Texas A&M University	1993
MCS	CS	Texas A&M University	1988
BS	CS	Texas A&M University	1986

Academic Experience

Institution	Rank	Title	When	Fulltime/part time
University of Alaska Fairbanks	Assoc	Prof	2001-Pres	Full
San Diego Supercomputer Center		Staff Scientist	1996-2001	Full
University of Alaska Fairbanks	Asst	Prof	1993-1996	Full

Non -academic Experience

Entity	Title	Brief Description of Position	When	Fulltime/part time

Certifications or Professional registrations

Current memberships in professional organizations

ACM

Honors and awards

- IEEE Visualization 2004, Panel Organizer, "What Should We Teach in a Scientific Visualization Class?", Best Panel Award.

- "Volume Visualization of the Orion Nebula" was awarded second place in the inaugural 2003 Science and Engineering Visualization Challenge (sponsored by NSF and Science Magazine). Three winners were chosen in each of three categories out of a total of 297 entries.

Service activities (within and outside the institution)

CEM Core Peer Review Committee (2010-2012)
CS Graduate Committee (2006-2011)
Science Advisory Committee for Hayden Planetarium (2008-09)
Faculty Senate (2004-06)
Faculty Senate President-Elect (2006-07)
Faculty Senate President (2007-08)
Faculty Alliance Member (2006-09)
CS Department Chair (2005-07)

Briefly list the most important publications and presentations from the past five years (title, co-authors if any, where published and/or presented, date of publication or presentation)

3D Stereo Visualization of the Orion Nebula for the movie "Our Cosmic Origins" shown during the 2009 Summer Science Exhibition by The Royal Society (London)

"Solar-Terrestrial Interactions From Cosmic Collisions" shown in the 2007 SIGGRAPH Animation Theater (with Carter Emmart)

100-second aurora borealis visualization for the "Cosmic Collisions" Space Show at the Hayden Planetarium (opened 2006 in New York)

"MPIGlut: Powerwall Programming Made Easier", Journal of WSCG 2008 (with Orion Lawlor)

"Interactive Volume Rendering of Aurora on the GPU", Journal of WSCG 2010 (with Orion Lawlor)

Briefly list the most recent professional development activities

Colloquium for Information Systems Security Education, 2007-2009

Sabbatical to develop software to support the production of a new space show at the Hayden Planetarium (2008-2009)

NVAC Consortium Workshop (2005-07)

Kara L. Nance

EDUCATION: Ph.D. Computer Science, 1991, University of Oklahoma

ACADEMIC EXPERIENCE

2007 – Present: Chair and Professor, Department of Computer Science Department, ASSERT Center Director, University of Alaska Fairbanks

2001 – 2007: Professor, Department of Computer Science Department, ASSERT Center Director, University of Alaska Fairbanks

1997 – 2001: Associate Professor and Computer Science Program Coordinator, Advising Center Advisor, University of Alaska Fairbanks

1993 – 1997: Assistant Professor and Computer Science Program Coordinator, Advising Center Advisor, University of Alaska Fairbanks

1991 – 1993: Assistant Professor, Computer Science, Medaille College

INDUSTRY EXPERIENCE

1983 – Present NMI Private Contracting and Consulting, Systems Analysis and Development, Systems Security

2000 – 2001: Printing System Security, Software Engineer, IBM Printing Services, 1985 – 1987: Systems Analyst/Process Control, NASA Project, National Severe Storms Laboratory

CERTIFICATIONS

PMI Certified Project Management Professional, 2001, 2004, 2007, 2011

PROFESSIONAL AFFILIATIONS -

IEEE (Member), ACM (Senior Member), PMI (Certified PMP), HTCIA, (Member), Infragard (Member), ABET (Program Reviewer), ODNI (Senior Advisory Board)

SELECTED PUBLICATIONS (Past 5 years – 25 additional peer reviewed articles)

Nance, K., B. Hay, R. Dodge, A Seazzu, S. Burd. *Virtual Laboratory Environments: Methodologies for Educating Cybersecurity Researchers*. Journal of Methodological Innovations. Vol. 4, No 3. pp 3-14 (2009).

Nance, K. *Teach Them When They Aren't Looking: Introducing Security in CS1*. IEEE Security & Privacy. Volume 7, Issue 5. Sept. – Oct. 2009. Page(s): 53-55.

Nance, K. and B. Hay. *Computer Security-focused Programming Assignments in Foundational CS Courses*. Proceedings of the 13th Colloquium for Information Systems Security Education. Seattle, WA. June 2009. (Best Paper Award.)

Hay, B., M. Bishop, and K. Nance. *Live Analysis: Progress and Challenges*. IEEE Security & Privacy. Volume 7, Issue 2. March-April 2009. Page(s) 30-37.

Nance, K., M. Bishop, and B. Hay. *Virtual Machine Introspection: Observation or Interference?* IEEE Security & Privacy. Volume 6, Issue 5, Sept.-Oct. 2008 Page(s): 32 - 37

Hay, B., R. Dodge., and K. Nance. *Using Virtualization to Create and Deploy Computer Security Lab Exercises.* International Federation for Information Processing. 278, pp. 621-635. Boston: IFIP.

Hay, B. and Nance, K. 2008. *Forensics examination of volatile system data using virtual introspection*. SIGOPS Oper. Syst. Rev. 42, 3 (Apr. 2008), 74-82. DOI= http://doi.acm.org/10.1145/1368506.1368517

Amari, K., Gardner, R., Hay, B., Hay, Kerski, J., Nance, K., and A. Yasinsac, *Software Review and Security Analysis of the Diebold Voting Machine Software, Premier TSx Version 4.7.1 Supplement*, Security and Assurance in Information Technology Laboratory, Florida State University, December 19, 2007, for the Florida Department of State.

AWARDS AND HONORS: Change Magazine Young Leader Award**,** Golden Key Honor Society, UAF Outstanding Advisor Award (3 times)**,** Usibelli Outstanding Teaching Award Nominee (5 times)**,** Usibelli Outstanding Service Award Recipient **,** ASUAF Outstanding Faculty Member Award**,** Dennis Demmert

Award for Outstanding Service to Alaskan Native Students**.** NSF Presidential Award for Excellence in Science, Mathematics, and Engineering Mentoring Nominee**,** Office of the Director of National Intelligence – Senior Advisory Board Member

SELECTED PRESENTATIONS (2010 – Spring Semester)

VENUE	TITLE	LOCATION	TYPE	DATE(S)
Hawaii International Conference on Systems Sciences	Digital Forensics Minitrack	Kauai, HI	Minitrack Chair	01/09/10
Digital Forensics Minitrack: Hawaii International Conference on Systems Sciences	Developing a Research Agenda to Improve Digital Forensics Education	Kauai, HI	Paper Presentation	01/07/10
Software Service Minitrack: Hawaii International Conference on Systems Sciences	The PATH Project: Partnerships for Automated Transformations of Heterogeneous Datasets	Kauai, HI	Paper Presentation	01/06/10
Infragard	Protecting Corporate Digital Assets	Anchorage, AK	Invited Talk	01/22/10
Broadening Participation in Computing	Encouraging Alaska Native Participation in Computing	Marina del Rey, CA	Project Report	02/01/10
Office of the Director of National Intelligence Regional Colloquium	Security and Reliability: The Future	Honolulu, HI	Invited Talk	02/10/10 – 02/11/10
Office of the Director of National Intelligence Regional Colloquium	Security and Reliability: The Future	Atlanta, GA	Invited Talk	02/24/10
7th GENI Engineering Conference	A Methodology for Federation Frameworks/ Virtual Machine Introspection	Chapel Hill, NC	Demonstration	03/10
Honeynet Alliance Annual Meeting	Visualizations of Time-series Data for Honeypots	Mexico City, Mexico	Invited Talk	04/10
National Security Colloquium	Cyber Security: Future Threats	Virginia Tech	Invited Talk	06/04/10
Colloquium for Information System Security Education	Expanding the Educational Boundaries of CyberDefense Exercises	Baltimore, MD	Paper Presentation	06/10

Brian Hay

Education

Degree	Field	Institution	Date
PhD	Computer Science	Montana State University	Dec 2005
MS	Computer Science	University of Alaska Fairbanks	Dec 2001
BS	Computer Science	University of Alaska Fairbanks	Dec 2000

Academic Experience

Institution	Rank	Title	When	Fulltime/part time
UAF	Staff & Adjunct Faculty		2002-2006	Full time
UAF	Assistant Professor		2007-present	Full time

Non -academic Experience

Entity	Title	Brief Description of Position	When	Fulltime/part time
IBM	Software Engineer	Software development in a hardware group	2001-2002	Fulltime
NMI	Consultant	Data systems, security, virtualization, and digital forensics consulting	2001-present	Part-time

Current memberships in professional organizations

ACM, HTCIA, AAFS, Infragard

Honors and awards

- Best Paper at CISSE 2010
- Best Paper at HICSS 2010 E-Government Track

Service activities (within and outside the institution)

Digital Forensics for Law Enforcement Agencies
Reviewer/Program Committee for SADFE, HICSS, NeFX
Chair of Virtualization Minitrack at HICSS 2009-2011
Organizer of At-Large CCDC

Briefly list the most important publications and presentations from the past five years

Nance, K. and **B. Hay**. Expanding the Educational Boundaries of Cyber Defense Exercises. Proceedings of the 14th Colloquium for Information Systems Security Education. Baltimore, MD. June 2010.

Hay, B. Applications of Virtualization to Digital Forensics Education. Digital Forensics Track of 43rd Hawaii International Conference on Systems Sciences. January 2010. (**Mini-Track Best Paper**)

Hecker, C. and **B. Hay**. Securing E-Government Assets Through Automating Deployment of Honeynets for IDS Support. E-Government Track of 43rd Hawaii International Conference on Systems Sciences. January 2010. (**Best Paper Award**)

Nance, K. and **B. Hay**. Computer Security-focused Programming Assignments in Foundational CS Courses. Proceedings of the 13th Colloquium for Information Systems Security Education. Seattle, WA. June 2009. (**Best Paper Award**)

Nance, K., **Hay, B**., and M. Bishop. Investigating the Implications of Virtual Machine Introspection for Digital Forensics. ARES, pp.1024-1029, 2009 International Conference on Availability, Reliability and Security, March 2009

Hay, B., M. Bishop, and K. Nance, Live Analysis: Progress and Challenges, IEEE Security and Privacy, vol. 7, no. 2, pp. 30-37, Mar./Apr. 2009, doi:10.1109/MSP.2009.43

Nance, K., **B. Hay**, and M. Bishop. Digital Forensics: Defining a Research Agenda. Proceedings of the 2009 HICSS Digital Forensics Minitrack. January 2009.

Nance, K., M. Bishop, and **B. Hay** Virtual Machine Introspection: Observation or Interference? IEEE Security and Privacy, vol. 6, no. 5, pp. 32-37, Sep./Oct. 2008, doi:10.1109/MSP.2008.134

Clarkson, M., **B. Hay**, M. Inge, A. Shelat, D. Wagner, and A. Yasinsac. Software Review and Security Analysis of the Scytl Remote Voting Software. Prepared for the Florida Department of State. Tallahassee, FL. August 2008.

Hay, B. and Nance, K. 2008. Forensic examination of volatile system data using virtual introspection. SIGOPS Oper. Syst. Rev. 42, 3 (Apr. 2008), 74-82. DOI= http://doi.acm.org/10.1145/1368506.1368517

Amari, K., Gardner, R., **Hay, B**., Kerski, J., Nance, K., and A. Yasinsac, Software Review and Security Analysis of the Diebold Voting Machine Software, Premier TSx Version 4.7.1 Supplement, Security and Assurance in Information Technology Laboratory, Florida State University, December 19, 2007, for the Florida Department of State.

I also published an additional 15 other papers during the past 5 years (2006-2011), and presented my work in multiple academic and non-academic settings.

Briefly list the most recent professional development activities

Attendance and participation in multiple conferences, including CISSE, HICSS, NeFX, several GECs, SANS events, Infragard meetings, and AAFS.

Orion Sky Lawlor

Education – degree, discipline, institution, year

Degree	Field	Institution	Date
BS	CS	U. Alaska Fairbanks	1999
BS	Math	U. Alaska Fairbanks	1999
MS	CS	U. Illinois Urbana-Champaign	2001
PhD	CS	U. Illinois Urbana-Champaign	2005

Academic Experience

Institution	Rank	Title	When	Fulltime/part time
U. Illinois Urbana-Champaign Center for Simulation of Advanced Rockets	Visiting Scholar		2001-2005	Fulltime
U. Alaska Fairbanks	Assistant Professor		2005-present	Fulltime

Non -academic Experience

Entity	Title	Brief Description of Position	When	Fulltime/part time
IBM	Intern	Summer Internship at T.J. Watson	1999	Fulltime

Certifications or Professional registrations

- N/A

Current memberships in professional organizations

- Member, ACM

Honors and awards

- 2006 Advisor for two MCM Meritorious Winner UAF teams
- 2007 Faculty Merit Bonus
- 2008 Advisor for MCM Outstanding Winner and INFORMS prize winning UAF team
- 2008 Advisor for MATE ROV Inspiration for Future Engineers award-winning UAF team
- 2009-2010 Outstanding Teaching Award for Computer Science Service activities (within and outside the institution)
- 2007-2008: Assisted federal prosecutor in analyzing digital forensic evidence
- 2005-2010: Contributions to several open-source software projects
- 2010 Chair, UAF Research Advisory Committee
- 2009-2010 Member, UAF Graduate Academic and Advisory Committee

- 2009-2010 Member, UAF Technology Advisory Board
- 2009-2010 UAF Faculty Senate

Briefly list the most important publications and presentations from the past five years (title, co-authors if any, where published and/or presented, date of publication or presentation)

- Message passing for GPGPU clusters: cudaMPI.
  Orion Sky Lawlor.
  In IEEE Cluster PPAC Workshop, 2009-08.
- MPIglut: Powerwall programming made easier.
  Orion Sky Lawlor, Matthew Page, and Jon Genetti.
  Journal of WSCG, pages 130-137, 2008-02.
- Interpolation-friendly soft shadow maps.
  Orion Sky Lawlor.
  In Proceedings of Computer Graphics and Virtual Reality '06
- Multiple flows of control in migratable parallel programs.
  Gengbin Zheng, Orion Sky Lawlor, and Laxmikant V. Kale.
  In Proceedings of 8th Workshop on High Performance Scientific and Engineering Computing (HPSEC-06). IEEE Press, August 2006.
- ParFUM: A parallel framework for unstructured meshes for scalable dynamic physics applications.
  Orion Sky Lawlor, Sayantan Chakravorty, Terry L. Wilmarth, Nilesh Choudhury, Isaac Dooley, Gengbin Zheng, and Laxmikant V. Kale.
  Engineering With Computers, 22(3):215-235, 2006.
- A system integration framework for coupled multiphysics simulations.
  Xiangmin Jiao, Gengbin Zheng, Phillip A. Alexander, Michael T. Campbell, Orion S. Lawlor, John Norris, Andreas Haselbacher, and Michael T. Heath.
  Engineering with Computers, 22(3-4):293-309, 2006.
- Performance degradation in the presence of subnormal floating-point values.
  Orion Sky Lawlor, Hari Govind, Isaac Dooley, Michael Breitenfeld, and Laxmikant Kale.
  In Proceedings of the International Workshop on Operating System Interference in High Performance Applications, September 2005.
- An integration framework for simulations of solid rocket motors.
  Xiangmin Jiao, Gengbin Zheng, Orion Lawlor, Phil Alexander, Mike Campbell, Michael Heath, and Robert Fiedler.
  In 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Tucson, Arizona, July 2005.

Briefly list the most recent professional development activities

- UIUC NASA Grant Meeting, 2008-07
- IEEE Cluster conference, 2009-08
- ORNL Invited visit, 2009-10
- ARSC Symposium, 2010-02

CHRIS HARTMAN

Education:

Degree	Field	Institution	Date
BS	CS	U. Alaska Fairbanks	1991
BS	Math	U. Alaska Fairbanks	1991
PhD	Math	U. Illinois	1997

Academic Experience:
UAF Assistant Professor 1997-2002
UAF Associate Professor 2002-present

Non-academic experience:
NCSA Research Programmer (Computer graphics) 1994-1996, part time

Certifications or Professional registrations:
None

Current memberships in professional organizations
None

Honors and awards
None

Service activities (within and outside the institution)
- Honors Faculty Advisory Council
- Referee/reviewer for Discrete Mathematics, Acta Mathematica, and other mathematics journals

Most important publications over last five years:

Glenn G. Chappell, John Gimbel, and Chris Hartman, Thresholds for Path Colorings of Planar Graphs, in Topics in Discrete Mathematics (M. Klazar, J. Kratochvíl, M. Loebl, J. Matousek, R. Thomas, P. Valtr, eds.), Springer, Berlin, 2006, pp. 435–454.

Chappell, G. G., Gimbel, J., C. Hartman. Bounds on the metric and partition dimensions of a graph, in Ars Combinatoria 88, July 2008, pp. 349–366.

Long, J.,C. Hartman. ODES: an overlapping dense sub-graph algorithm,
in Bioinformatics 2010 Nov 1;26(21):2788-9. Epub 2010 Sep 9.

Most recent professional development activities
- Independent study of recent developments in software design, design patterns, and agile programming practices.
- Attendance at relevant seminars at UAF

Glenn G. Chappell

Education

- Ph.D. in mathematics, University of Illinois, 1996. Advisor: Douglas B. West.
  Thesis: Optimization on Products of Combinatorial Structures.
- M.A. in mathematics, University of Kansas, 1990. Advisor: Jack Porter.
- B.S. in mathematics with concentration in computer science, University of Kansas, 1988.

Experience

Associate Professor of Computer Science

University of Alaska Fairbanks, Fairbanks, Alaska Fall 2006–present

- Teach computer sciences courses including Data Structures and Algorithms, Computer Science I Computer Science II, and Automata & Formal Languages.
- Organize a faculty research group in graph theory.

Assistant Professor of Computer Science

University of Alaska Fairbanks, Fairbanks, Alaska Fall 2000–Spring 2006

- Taught computer sciences courses including Computer Science I, Unix, Computer Graphics, Advanced Computer Graphics, Topics in Computer Graphics, Data Structures and Algorithms.
- On staff at the Arctic Region Supercomputing Center. Did research in computer graphics, virtual reality, user-interface design.
- Organized a faculty research group in graph theory.

Assistant Professor of Mathematics

Southeast Missouri State University, Cape Girardeau, Missouri Fall11996–Spring 2000

- Taught mathematics courses including Discrete Mathematics, Linear Algebra, Numerical Analysis, Calculus, Applied Calculus, Trigonometry.

Teaching Assistant/Fellow

University of Illinois, Urbana, Illinois Fall 1990–Summer 1996

- Taught mathematics courses including Calculus, Calculus for Social Scientists, discussion sections of Computers for Elementary School Teachers.
- Taught in the Calculus&*Mathematica* program, as well as in the traditional classroom.
- Assisted a professor in teaching a freshman honors tutorial in geometric computer graphics. Designed and wrote tutorials on cellular automata.
- Assisted a professor in teaching a summer computer lab course on conic sections for junior high school teachers. Helped design and write course software.

Other Positions and Fellowships

- Contributing editor, *American Mathematical Monthly* problems column, 1998–2002.
- Awarded U.S. Department of Education National Needs Fellowship, University of Illinois, 1994-95 and 1995-96.

Appendix F – Table of Substitutions

TABLE OF SUBSTITUTIONS
2010-2011 Course Catalog

Baccalaureate Core Requirements (number of credits needed)	To meet these UAF core course requirements:	Use any of these UAA general education courses:	Use any of these UAS general education courses:
COMMUNICATION (9 cr)			
Written Communication (3 cr)	ENGL F111X	ENGL 111	ENGL 111
Written Communication (3 cr)	ENGL F211X or F213X	ENGL 211, 212, 213, 214, 311, 312 or 414	ENGL 211, 212
Oral Communication (3 cr)	COMM F131X or F141X	COMM 111, 235, 237 or 241	COMM 111, 235, 237, 241
PERSPECTIVES ON THE HUMAN CONDITION (18 cr)*			
History (3 cr) Political Economy (3 cr) Social Culture (3 cr)	HIST F100X ECON/PS F100X ANTH/SOC F100X	ANTH 101, 200, 202, 250; CEL 292; BA 151; ECON 201, 202, 210; EDEC 105; ENVI 212; GEOG/INTL 101; HNRS 292; HS 220; HUMS/SWK 106; INTL 101; JPC 101; JUST 110, JUST/SOC 251, JUST 330; LSSS 111; PARL 101; PS 101, 102, 311, PS/SOC 351; PSY 111, 150; SOC 101, 110, 201, 202; SWK 243; WS200	ANTH 101, 202, 211 ECON 100, 201, 202 GEOG 101; GOVT 101, 102, 230, 251; HIST 105, 106, 131, 132; PSY 101, 250; SOC 101, 201
Literature (3 cr)	ENGL/FL F200X	ART 261, 262, 36OA, 360B; ENGL 121, 201, 202, 301, 302, 305, 306, 307, 310, 383, 445; HIST 101, 102, 121, 122, 131,132, 341; HNRS 192; HUM 211, 212, 250; (Languages: AKNS 101A, 101B, 101C, 102A, 102B, 102C, 201; CHIN 101, 102; ASL, FREN, GER, JPN, RUSS, SPAN 101, 102, 201, 202); LING 101; MUS 221, 222; PHIL 101, 201, 211, 212, 313, 314; PS 331, 332, 333; THR 311, 312, 411, 412	ENGL 215, 223, 224, 225, 226, 261; HIST 105, 106, 131, 132; HUM 120; JOUR 101; (Languages: AKL 105, 106, 107, 108; ASL, FREN, SPAN, RUSS 101, 102 or other approved world languages); PHIL 101, 201, 271
Aesthetics (3 cr)	ART/MUS/THR F200X, or HUM F201X, or ANS F202X	AKNS/MUS 215; ART 160, 261, 262, 360A, 360B; DNCE 170; MUS 121, 124, 221, 222; THR111, 311, 312, 411, 412	ART 160, 261, 262; MUS 123; THR 111, 211, 212
Ethics (3 cr)	BA F323X, or COMM F300X, or JUST F300X, or NRM F303X, or PS F300X, or PHIL F322X	PHIL 301, 302, 303, 304, 405	PHIL 301
FOREIGN LANGUAGE OPTION			
*OR complete 12 cr from the Perspectives on the Human Condition options above, plus 2 semester-length courses in a single Alaska Native language or other non-English language or 3 semester-length (9 cr) in American Sign Language at the university level.			
MATHEMATICS (3 cr)			
	MATH 103X, F107X, F161X, F200X, F201X, F202X, F262X, or F272X; STAT F200X, or any math course having one of these as a prerequisite	MATH 107, 108, 109, 172, 200, 201, 272; STAT 252, 253, 307	MATH 107, 131 (or higher mathematics course for bachelor's degree); STAT 107 (or higher statistics course for bachelor's degree)
NATURAL SCIENCES (8 Credits)			

Baccalaureate Core Requirements (number of credits needed)	To meet these UAF core course requirements:	Use any of these UAA general education courses:	Use any of these UAS general education courses:
Complete any two 4-credit natural science courses with lab sections	ATM F101X BIOL F100X, F103X, F104X, F111X, F112X, F115X, F116X CHEM F100X, F103X, F104X, F105X, F106X GEOG F211X GEOS F100X, F101X, F112X, F120X, F125X MSL F111X PHYS F102X, F103X, F104X, F115X, F116X, F175X, F211X, F212X, F213X	ASTR 103, 104; BIOL 102 & 103, 111, 112, 115, 116, 178 & 179; CHEM 103 & 103L, 104 & 104L, 105 & 105L, 106 &106L; ENVI/GEOG 211 & 211L; GEOL 111, 115 & 115L, 178 & 179; LSIS 102, 201, 202; PHYS 123 & 123L, 124 & 124L, 211 & 211L, 212 & 212L (must include at least 1 credit of lab to meet UAF core requirement)	BIOL 103, 104, 105, 106, 111, 112; CHEM 103, 105, 106; ENVS 101; GEOL 104; PHYS 102, 103, 104, 211, 212 (must include at least 1 credit of lab to meet UAF core requirement)

Appendix G – Advising Workshop Notes

Advising Workshop Report – 04/02/2010

Attendance	Approximately 38 students attended (included majors, 4 graduate students, and non degree) as well as the following: Glenn Chappell (CS Faculty) Jon Genetti (CS Faculty) Chris Hartman (CS Faculty) Orion Lawlor (CS Faculty)
Results	• 15 students completed and turned in their registration forms • Additional students started their forms • 2 new students attended.
Supplies on hand	All CS degree majors' registration forms printed and available to students 15 pizzas $160.00 2 cases of soda $15.00 2 cases bottled water $8.00 4 dozen cookies $24.00 30 Undergraduate degree audit forms 50 Schedule worksheets 50 Fall 2010 Schedules 5 2009-2010 UAF Catalogs 30 Four-year plans 50 CS 2009-10 degree check sheet 10 Change of major forms 30 Projected course offering/prerequisite structures All degree majors printed registration forms available to students Computer lab across the hall available for student for UAonline access.
Announcements	Introductions New student information Changing your major Plan through to graduation Junior/senior information Degree audit Applying for graduation M.S. program Graduate students Graduate Study Plan Job Opportunities Schedule Changes 1. Instructors described senior level electives (CS 481, CS 425, CS 463) 2. Warning Math 307 offered for FALL 2010 only – no spring class. Graduating seniors need to take Math 307 in the Fall 3. Warning about conflict between PHYS 212, CS 321, and MATH 314. Plan carefully to avoid this conflict which occurs every spring. Advisor balancing Registration dates: Monday, April 5 – Graduates, Seniors, Juniors, Sophmores, Freshman Wednesday, April 6 – Continuing non-degree

Advising Workshop Report – 11/04/2010

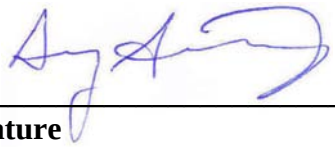
Attendance	Approximately 25 students attended (21 included majors, 6 graduate students, and non-degree) as well as the following: Glenn Chappell (CS Faculty) Jon Genetti (CS Faculty) Chris Hartman (CS Faculty) Orion Lawlor (CS Faculty)
Results	• 15 students completed and turned in their registration forms • Additional students started their forms • 2 new students attended.
Supplies on hand	All CS degree majors' registration forms printed and available to students 15 pizzas $160.00 2 cases of soda $15.00 2 cases bottled water $8.00 30 Undergraduate degree audit forms 50 Schedule worksheets 50 Spring 2011 Schedules 2 2010-2011 UAF Catalogs 30 Four-year plans 50 CS 2010-2011 degree check sheet 10 Change of major forms 30 Projected course offering/prerequisite structures All degree majors printed registration forms available to students Computer lab across the hall available for student for UAonline access.
Announcements	Introductions New student information Changing your major Plan through to graduation Junior/senior information Degree audit Applying for graduation M.S. program Graduate students Graduate Study Plan Job Opportunities Schedule Changes 1. Instructors described senior level electives 2. Possible addition to the schedule of Math 307 for the Spring semester instructed by Dr. Hartman 3. Warning about conflict between PHYS 212, CS 321, and MATH 314. Plan carefully to avoid this conflict which occurs every spring. Advisor balancing Registration dates: Monday, November 8 – Graduates, Seniors, Juniors, Sophomores, Freshman Wednesday, November 10 – Continuing non-degree

Signature Attesting to Compliance

By signing below, I attest to the following:

That the **B.S. Computer Science Program** has conducted an honest assessment of compliance and has provided a complete and accurate disclosure of timely information regarding compliance with ABET's *Criteria for Accrediting Computing Programs* to include the General Criteria and any applicable Program Criteria, and the ABET *Accreditation Policy and Procedure Manual.*

Douglas J. Goering
Dean's Name (As indicated on the RFE)



Signature

June 24, 2011
Date