SELF-STUDY QUESTIONNAIRE
FOR THE REVIEW OF

Software Engineering

Prepared for:

Engineering Accreditation Commission
Accreditation Board for Engineering and Technology
111 Market Place, Suite 1050
Baltimore, Maryland 21202-4012
Phone: 410-347-7700 Fax: 410-625-2238
e-mail: eac@abet.org www: http://www.abet.org/

Prepared by:

Department of Computer Science and Software Engineering
Rose-Hulman Institute of Technology

June 9, 2006

ROSE-HULMAN INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING
5500 WABASH AVENUE, CM 97 TERRE HAUTE, INDIANA 47803-3999
TELEPHONE: (812) 877-8402 FAX: (812) 872-6060 EMAIL: Don.Bagert@rose-hulman.edu

Table of Contents

Table of Contents (continued)

Self-Study Report

for Software Engineering

A. Background Information

1. Degree Titles

Bachelor of Science in Software Engineering

2. Program Modes

The software engineering program is offered on-campus for full-time students. All classes are conducted on campus and are taught in face-to-face mode. Students may do co-op work or internships, but neither is required for graduation.

3. Actions to Correct Previous Shortcomings

This section is not applicable, since this is the first accreditation review of the software engineering program.

4. Contact Information

Dr. Cary Laxer, Head
Department of Computer Science and Software Engineering
Rose-Hulman Institute of Technology
5500 Wabash Avenue, CM 100
Terre Haute IN 47803-3999
812-877-8429
laxer@rose-hulman.edu

Dr. Donald J. Bagert, PE
Director of Software Engineering & Software Engineering Accreditation Coordinator
Rose-Hulman Institute of Technology
5500 Wabash Avenue, CM 97
Terre Haute IN 47803-3999
Don.Bagert@rose-hulman.edu

B. Accreditation Summary

1. Students

a. Evaluation

Faculty members are primary evaluators of students in the software engineering (SE) program. Seventy-two of the 192 hours (37.5%) required for the SE degree are taught within the Department of Computer Science and Software Engineering (CSSE). Each of those courses contain a set of outcomes, each of which is related to at least one of the software engineering program outcomes described in Section B.3. Each course also has a course assessment plan (CAP), which is used to determine whether or not students have met the stated course outcomes. Each course outcome is assessed in at least two ways as specified in the CAP, e.g. through project work, examinations, homework assignments, and laboratory assignments.

For courses not taught in the CSSE department, the program depends on the professional judgment of the faculty in Mathematics (MA), Electrical and Computer Engineering (ECE), Humanities and Social Sciences (HSS) and other departments within the Institute.

b. Advising

All incoming freshmen at Rose-Hulman are assigned a freshman academic advisor, which may be any member of the Rose-Hulman faculty. (However, over the first three years of the SE program, over 80% of the first-year software engineering students have had a CSSE faculty member as a freshman advisor.) There are about 30 freshman advisors (all volunteers), with most of them having approximately 15 freshmen advisees. In recent years, CSSE faculty members have served as freshman advisors for first-year computer science and software engineering majors.

After the first three terms, all software engineering majors are assigned a permanent advisor from among the CSSE faculty. Software engineering majors transferring from other colleges are also assigned a faculty member in CSSE as their permanent advisor upon entering the Institute. All CSSE faculty act as permanent or freshman advisors; there is approximately a 16:1 ratio of advisee to permanent advisor.

Before registration for each term, every software engineering major, regardless of classification, must meet with his or her advisor and receive counsel on what courses to take for the next term. At the end of this meeting, the advisor gives the student a personal identification number (PIN) which can subsequently be used for online registration.

c. Monitoring

Each academic advisor has online access to midterm and final grades for all their advisees. Midterm grades are posted at the end of the fifth week of the 10-week term and are made available to both student and advisor, who together subsequently discuss the midterm grades during a meeting which usually that next week.

In order to remain in good academic standing, a student with at least four terms at Rose-Hulman must maintain a 2.0 grade point average (GPA) both for the current term and cumulatively over all terms. (In terms 1-3, this GPA requirement is 1.8.) If a student previously in good standing fails to meet this requirement, the student is placed on academic probation and is sent a letter

with both the details of probation and what must be done in the subsequent term in order to return to good standing. A copy of this letter is sent to the student's academic advisor, who can then take appropriate action, usually in the form of one or more meetings with the advisee.

Most Rose-Hulman undergraduate students complete graduation requirements in the standard four years of study. Advisor and advisee work together to plan those four years of study, taking into account any double majors, minors or certifications. Each student is strongly encouraged to maintain a term-by-term degree plan in order to minimize the possibility of not being able to take all courses in a timely manner for graduation. Also, at the beginning of the senior year, each student (except for double majors) receives a spreadsheet from the Office of the Registrar, outlining what requirements still need to be met in order to graduate. Double majors have to file a plan of study at the beginning of their junior year, which has to be approved by both departments and the institute's Curriculum Committee.

Registrar's Office

The Office of the Registrar, under the direction of Mr. Tim Prickel, is the primary repository of student academic records. The Registrar and the Associate Registrar, Ms. Jan Lind, ensure that student records are both current and correct. A file is maintained for each student. The file contains the mid-term and final grade reports for each academic quarter; these reports show how the student is performing in each course. The file also contains other pertinent academic information, including memoranda from the Registrar regarding academic warnings, academic probation, and/or failure to maintain the GPA-in-major requirement. There are also program checklists so the Registrar can track when a student fulfills each requirement (both for the department and for the Institute). The student file maintained by the Registrar's Office is the definitive record that charts a student's progress through the four-year program. The Office of the Registrar follows the Institutional Academic Rules and Procedures Manual; this manual is available to students both on line and in printed form from the Office of Student Affairs.

In order to monitor student completion of academic requirements, the following procedure is used. During the freshman, sophomore, and junior years, the primary responsibility for tracking student progress lies with the student's Freshman Advisor (first year) or Academic Advisor (subsequent years). The advisor keeps a Program Checklist in order to track the student's completion of required courses for the major, as well as all Institute graduation requirements. Prior to the fall of the student's senior year, the Registrar's Office reviews the file of each rising senior to confirm that the student is on track for graduation. If the student has failed to take a required course or to fulfill the appropriate number and/or distribution of Humanities and Social Sciences courses, the student is notified by the Registrar's Office. This gives the student three quarters by which to complete the necessary courses and requirements. Through the senior year, the student is tracked by the Registrar's Office until the Final Degree Audit, a process by which all students' requirements are checked and verified. Finally the entire faculty approves all students for graduation during a meeting of the faculty. At that time, all students are approved by a vote of the faculty.

The Office of the Registrar provides other services for faculty and students at Rose-Hulman. The office has an excellent track record in arranging the schedule of course offerings so that courses required for a given program will not be offered at conflicting times. The Office also ensures that students who must take a required course at a particular time are accommodated

before students whose schedules are a bit more flexible. In addition, the staff of the office provides support to the academic advisors. During the Freshman Advisor Orientation, the Registrar gives a presentation familiarizing advisors with new or existing policies. Guidance is also offered on an individual basis to faculty who require more information on how to access students schedule information, rules and procedures, and on-line course registration.

d. Acceptance of Transfer Students

RHIT has established a policy for the admission of transfer students to the Institute. Our department follows these policies. In order to be admitted as a transfer student to RHIT, a student must be in good academic standing at his/her current college or university.

To date, there have been only five SE transfer students, two in Fall 2004 and three in Fall 2005. The first of these students are scheduled to graduate in May 2007. All five of these transfers have over a 3.0 (out of 4) cumulative GPA at Rose-Hulman and are expected to graduate on time.

e. Validation of Transfer Credit

RHIT establishes policies for transfer credit at the Institute level. Our department follows these policies. A transfer student may be given credit for equivalent courses taken elsewhere if a grade of "C" or better was earned, and the course is at the 100-level or above. The head of the department in which the courses would have been taught evaluates both courses and credits. This is done on a case-by-case basis. Transfer of engineering courses taken elsewhere can only occur if the student was enrolled in a program accredited by ABET; transfer of other courses (such as humanities and social sciences) can only occur it the students was enrolled in a college or university that is accredited by a nationally recognized accrediting agency, such as the North Central Association.

Usually the granting of transfer credit is a smooth process, since students often transfer in from ranked and recognized colleges and universities. If the school is not well known, however, transfer credits will only be approved after the appropriate department chairman has talked to the student about the course(s) being transferred, looked over the catalog description(s), and determined what text was used in the course(s) and how much of the book was covered. Final credit transfers are not made until it is determined that the student can progress satisfactorily using this prior background information. Final acceptance of the credit is at the discretion of the engineering department head.

Summer Courses Transferred to RHIT

Students planning on taking courses during the summer at another institution and transferring them back to Rose-Hulman must secure prior approval from the department that the proposed course is acceptable and meets all requirements. The monitoring of this program is the responsibility of the academic advisor and the appropriate department chair. The overall monitoring of the transfer credit program is the responsibility of the department chairman and the Registrar. The Registrar assures performance and consistency among all academic departments.

Software Engineering Transfer Students

To date, there have been only five SE transfer students, two in Fall 2004 and three in Fall 2005.

All five of these transfers have over a 3.0 (out of 4) cumulative GPA at Rose-Hulman. The advisor for all of these students feels that they were placed in the proper courses for their first term at Rose-Hulman, and the advisor is satisfied with their progress after transferring to Rose. (One student did have one course of transfer credit added shortly after arrival on campus.)

2. Program Educational Objectives

Software engineering graduates will

PEO1. develop complex systems (including analysis, design, construction, maintenance, quality assurance and project management) using the appropriate theory, principles, tools and processes.

PEO2. use appropriate computer science and mathematics principles in the development of software systems.

PEO3. solve problems in a team environment through effective use of written and oral communication skills.

PEO4. have knowledge of current issues presently involved in effectively performing duties as a software practitioner in an ethical and professional manner for the benefit of society.

PEO5. practice the lifelong learning needed in order to keep current as new issues emerge.

PEO6. develop software in at least one application domain.

These objectives are consistent with the mission of the institution and the constituents of the program.

a. Mission

The mission of the Rose-Hulman Institute of Technology to provide students with the world's best undergraduate education in engineering, mathematics, and science in an environment of individual attention and support. With that in mind, the software engineering program provides its graduates with the best possible preparation to be effective software professionals, in order to be able to achieve the Program Educational Objectives within 3-5 years of graduation.

b. Program Constituencies

Program constituencies (stakeholders) include

- Current students,
- Employers of software engineering graduates, and
- Alumni of the program.

The list of program constituencies was identified by the CSSE faculty and reviewed by the Board of Advisors for the Department of Computer Science and Software Engineering. The CSSE Board of Advisors represents two of these three constituencies: CSSE alumni and employers of CSSE graduates. Thus, the CSSE Board of Advisors is an ongoing, critical source for constituent feedback.

c. Process for Establishment and Review of the Program Educational Objectives

The CSSE department has implemented a Continuous Improvement Process (CIP) which includes both Computer Science and Software Engineering baccalaureate programs as well as all courses taught within the department. The components of the software engineering program undergo continuous improvement through the CIP include

- The software engineering Program Educational Objectives,
- The software engineering curriculum, and
- The assessment methodology for the entire software engineering program.

The software engineering Program Educational Objectives were developed by the CSSE faculty (during the 2002-03 academic year, as part of development of the proposal for SE program), after receiving input from the CSSE Board of Advisors. Table B-1 provides a timeline of the development, review and modification of the Program Education Objectives to date.

Table B-1. Timeline for development, review and modification of the Program Educational Objectives to date.

Date	**Activities**
October 2002	• Input received on potential Program Educational Objectives by CSSE Board of Advisors
January 2003	• SE Program Educational Objectives approved by CSSE faculty
May 2003	• SE Program Educational Objectives reviewed by CSSE Board of Advisors
October 2003	• SE Program Educational Objectives reviewed by CSSE Board of Advisors
October 2004	• SE Program Educational Objectives reviewed by CSSE Board of Advisors • SE Program Educational Objectives reviewed by CSSE faculty
August 2005	• SE Program Educational Objectives reviewed by CSSE faculty
October 2005	• SE Program Educational Objectives reviewed by SE mock accreditation reviewer – change in tense suggested to make it clear that these are for graduates 3-5 years in the future
March 2006	• SE Program Educational Objectives reviewed by CSSE Board of Advisors – suggested separating one objective into what is now • Recommended revisions from mock review and Board of Advisors approved by the CSSE faculty

d. Process to Ensure Achievement of Program Educational Objectives

The portion of the CIP relevant to Program Educational Objectives can be found in Figure B-1. Each year a Program Assessment Plan (PAP) is developed by the Software Engineering Program Coordinator and approved by the CSSE department faculty. Part of the purpose of the PAP is to evaluate the Program Educational Objectives for the software engineering program.

The PAP for the 2005-06 academic year can be found in Appendix III.A. It specifies two means of assessment of the program:

- Alumni Survey Results
- Board of Advisors Meeting Minutes

The Alumni Survey Results are derived from an annual survey of all alumni of the program, conducted by the Institute, with specific questions asked to graduates of a particular major. Success on the alumni survey for a particular program objective is that at least 60% of the SE alumni who graduated 3-5 years ago responding the survey feel that they have met that objective.

The Board of Advisors also reviews the Program Educational Objectives at their annual meeting, recommends changes if necessary, and records the result in their meeting minutes.

Although students are constituents of the program, they are not included in the assessment of the Program Educational Objectives, since said objectives are looking 3-5 years after graduation.

The data from the Alumni Survey Results and Board of Advisors Minutes is then incorporated into a Program Assessment Report, which is sent by the SE Program Coordinator to the CSSE faculty. The report is then reviewed CSSE faculty in mid-August, and make changes to the Program Educational Objectives if necessary.

At this time, there are no graduates of the SE program who have been in the workplace for at least three years (the first graduate of the degree program was in May 2005), so there is no way to measure achievement of the program educational objectives to date. Once there have been software engineering majors who have graduated at least three years ago (i.e. during the 2008-09 academic year), the PAP will then be used to assess whether the Program Educational Objectives are being achieved.

Table B-2 contains the projected timeline for assessment starting in 2008-09.

e. Relationship of Program Educational Objectives to ABET Program Outcomes

The relationship of the software engineering Program Educational Objectives to the ABET Program Outcomes (a) through (k) of EAC Criterion 3 can be found in Table B-3.

Figure B-1. Process for ongoing assessment and improvement of SE Program Educational Objectives.

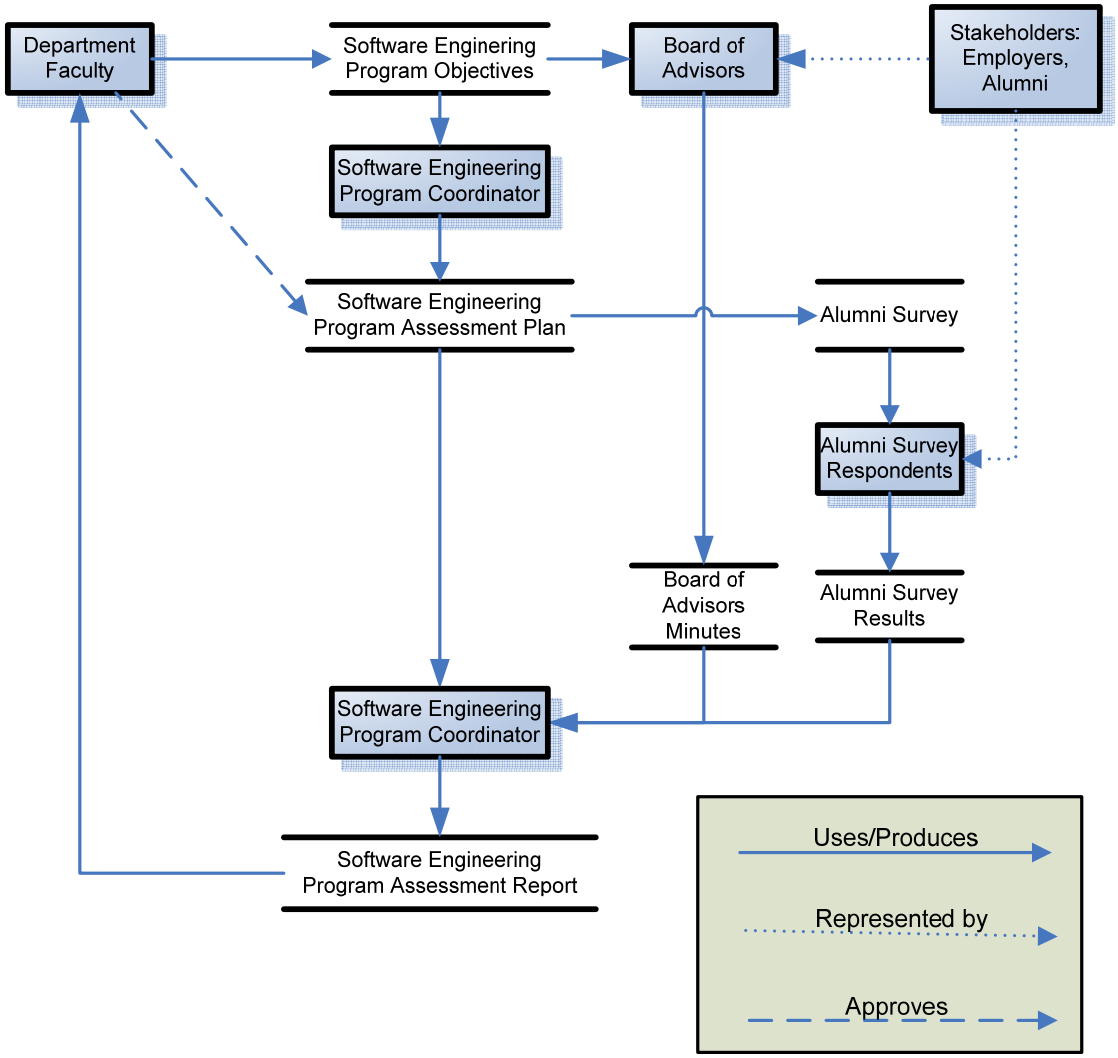


Table B-2. Timeline for assessment and improvement of Program Educational Objectives over a year.

Time of Year	**Activities**
August	• CSSE Faculty approve any change in SE Program Educational Objectives • SE Program Coordinator develops Program Assessment Plan for next year • CSSE faculty approve new SE Program Assessment Plan
September through May	• Software Engineering Alumni are surveyed on SE Program Educational Objectives • CSSE Board of Advisors review SE Program Educational Objectives
Summer	• SE Program Coordinator submits Program Assessment Report to CSSE faculty • Program Assessment Report approved by CSSE faculty

Table B-3. ABET a-k outcomes mapped to software engineering Program Educational Objectives.

ABET Criterion 3 Program Outcomes		**Software Engineering Program Educational Objectives**					
		Develop Complex Systems	Use Appropriate CS and Math	Team Env. Using Comm. Skills	Ethics and Profession.	Lifelong Learning	Appl. Domain
		PEO1	**PEO2**	**PEO3**	**PEO4**	**PEO5**	**PEO6**
Applying math, science & engineering	**a**	✓	✓				
Experiment & interpret data	**b**	✓	✓				
Design	**c**	✓					
Teams	**d**	✓		✓			
Problem solving	**e**	✓		✓			
Professional & ethical responsibility	**f**				✓		
Communications	**g**			✓			
Global society	**h**				✓		
Lifelong learning	**i**					✓	
Contemporary issues	**j**				✓		
Techniques, skills, tools	**k**	✓					✓

3. Program Outcomes and Assessment

a. Program Outcomes

By graduation:

Software engineering graduates will have demonstrated	
SE1.	the ability to apply software engineering theory, principles, tools and processes, as well as the theory and principles of computer science and mathematics, to the development and maintenance of complex software systems.
SE2.	the ability to design and experiment with software prototypes.
SE3.	the ability to select and use software metrics.
SE4.	the ability to participate productively on software project teams involving students from both software engineering and other majors.
SE5.	effective communication skills through oral and written reports and software documentation evaluated by both peers and faculty.
SE6.	the ability to elicit, analyze and specify software requirements through a productive working relationship with project stakeholders.
SE7.	the ability to evaluate the business and impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues.
SE8.	the ability to apply appropriate codes of ethics and professional conduct to the solution of software engineering problems.
SE9.	the knowledge required to understand the need for and the ability to perform in lifelong learning.
SE10.	the basic knowledge required in a software engineering application domain track.

The Program Outcomes foster attainment of the Program Educational Objectives discussed in Section B.2; Table B-3 provides a mapping.

Section B.3.b discusses the achievement of the software engineering program outcomes, while Section B.3.c explains how ABET Criterion 3 outcomes a-k are achieved through the RHIT Institutional Outcomes as implemented within the software engineering program.

b. Process to Ensure Achievement of Program Outcomes

The software engineering program outcomes were developed by the CSSE faculty (during the 2002-03 academic year, as part of development of the proposal for SE program), after receiving input from the CSSE Board of Advisors.

As stated in Section B.2, the CSSE department has implemented a Continuous Improvement Process (CIP) which includes both Computer Science and Software Engineering baccalaureate programs as well as all courses taught within the department. The components of the software engineering program that are related to program outcomes using the CIP include

- The software engineering curriculum,
- Individual courses in the CSSE department,

- The assessment methodology for the individual CSSE courses, and
- The assessment methodology for the entire software engineering program.

The portion of the CIP relevant to program outcomes can be found in Figure B-2. The Program Outcomes are determined by the CSSE departmental faculty. The Program Assessment Plan – developed by the Software Engineering Program Coordinator and approved by the CSSE faculty – is used to evaluate the Program Outcomes for the software engineering program (as well as its Program Educational Objectives, as discussed in Section B.2).

The Program Assessment Plan for the 2005-06 academic year (Appendix III.A), the first year a PAP was implemented, specifies three means of assessment of the Program Outcomes:

- Exit Interviews – used for all Program Outcomes
- Course Assessment Reports (CARs) – used for Program Outcomes SE1 through SE9
- Application Domain Curricular Requirements – used for Program Outcomes SE10

Exit Interviews

Each software engineering senior who is eligible to participate in the annual commencement exercises near the end of May (i.e. they are graduating or within 24 credit hours of doing so) is interviewed one-on-one by the Director of Software Engineering, usually sometime in May. Participation in an exit interview is encouraged but optional. As part of the interview, the students are asked to rate each Program Outcome using the following rubric:

Excellent - 5
Very Good - 4
Satisfactory - 3
Fair - 2
Poor – 1

Success for a Program Outcome in the exit interviews would mean that 80% or more of those interviewed expressed satisfaction when asked about that particular outcome. The results for 2005-06 can be found in Table B-4. Each of the outcomes exceeded the 80% satisfaction threshold, and 79 of the 80 responses were satisfactory or better.

Thus, the 2005-06 Program Assessment Report (PAR) for Software Engineering (Appendix III.B) lists all ten Program Outcomes as being met; however, it also makes the observation that the CSSE faculty discuss ways to improve outcomes SE3 (Software Metrics) and SE7 (Impact of Potential Solutions), each of which had an average response between 3 and 4.

Course Assessment Reports

The CSSE faculty determines the software engineering curriculum, which must be approved by the RHIT Curriculum Committee. Each CSSE course has a set of Course Outcomes, devised by the Course Coordinator (one of the CSSE faculty) and approved first by the Course Coordination Team (the coordinator and two other CSSE faculty) and then by the whole CSSE faculty.

Table B-3. Map of SE Program Outcomes to Program Educational Objectives.

Software Engineering Program Outcomes		**Software Engineering Program Educational Objectives**					
By graduation, students in the software engineering program will have		Develop Complex Systems	Use Appropriate CS and Math	Team Env./ Comm. Skills	Ethics and Profession.	Lifelong Learning	Appl. Domain
		PEO1	**PEO2**	**PEO3**	**PEO4**	**PEO5**	**PEO6**
the ability to apply software engineering theory, principles, tools and processes, as well as the theory and principles of computer science and mathematics, to the development and maintenance of complex software systems.	**SE1**	✓	✓				
the ability to design and experiment with software prototypes.	**SE2**	✓					
the ability to select and use software metrics.	**SE3**	✓	✓				
the ability to participate productively on software project teams involving students from both software engineering and other majors.	**SE4**			✓			
effective communication skills through oral and written reports and software documentation evaluated by both peers and faculty.	**SE5**			✓			
the ability to elicit, analyze and specify software requirements through a productive working relationship with project stakeholders.	**SE6**	✓		✓			
the ability to evaluate the business and impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues.	**SE7**	✓			✓		
the ability to apply appropriate codes of ethics and professional conduct to the solution of software engineering problems.	**SE8**	✓			✓		
the knowledge required to understand the need for and the ability to perform in lifelong learning.	**SE9**					✓	
the basic knowledge required in a software engineering application domain track.	**SE10**						✓

Figure B-2. Process for ongoing assessment and improvement of SE program outcomes.

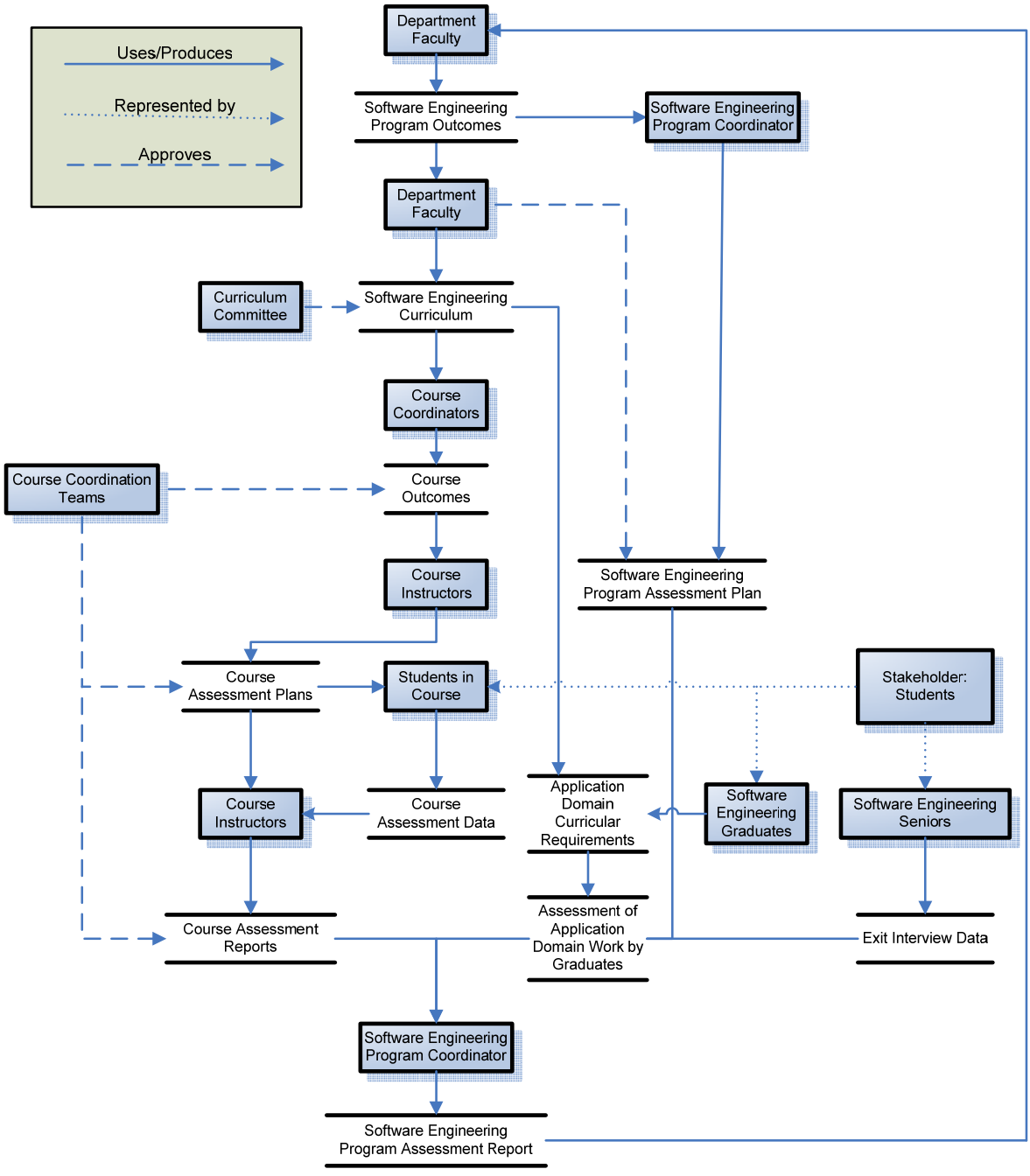


Table B-4. Exit interview responses, May 2006.

		Student Responses (with identities concealed)										
Program Outcome	**#**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**% ≥ 3**	**Success? (≥ 80%)**	**Average Response**
Apply SE, CS, Math to Software Systems	**SE1**	4	4	4	4	4	4	5	4	100%	**Yes**	4.13
Software Prototypes	**SE2**	4	3	3	5	4	4	5	3	100%	**Yes**	4.13
Software Metrics	**SE3**	4	3	3	4	4	3	4	4	100%	**Yes**	3.88
Teams	**SE4**	5	5	5	3	5	4	4	4	100%	**Yes**	4.25
Communication Skills	**SE5**	5	4	4	4	5	3	5	5	100%	**Yes**	4.38
Requirements. through Stakeholders	**SE6**	5	4	4	5	5	4	5	5	100%	**Yes**	4.75
Impact of Potential Solutions	**SE7**	5	4	4	3	2	3	4	3	87.5%	**Yes**	3.38
Ethics and Professional Conduct	**SE8**	5	5	5	4	3	4	5	4	100%	**Yes**	4.38
Lifelong Learning	**SE9**	4	5	5	5	4	5	4	4	100%	**Yes**	4.50
Application Domain	**SE10**	5	5	5	5	4	4	5	4	100%	**Yes**	4.63

The Course Outcomes must then be assessed for a particular term using a Course Assessment Plan (CAP) devised by the course instructor(s) for that particular term (and approved by the Course Coordination Team and entire CSSE faculty). The CAP specifies that each Course Outcome be assessed in two different ways using assessment tools such as homework or exam problems, and defines the success criteria for each assessment of each Course Outcome. A sample Course Assessment Plan can be found in Appendix III.C.

The implementation results for a particular CAP are summarized in a Course Assessment Report (CAR) compiled by the course instructor(s). If a particular Course Outcome is not met, the course instructor(s) include a recommendation for the next time the course is taught. Subsequently, the CAR is then again approved the Course Coordination Team and entire CSSE faculty. A sample Course Assessment Report can be found in Appendix III.D.

The Program Assessment Reports define success criteria for a particular program outcome from SE1 through SE9 using the CARs as success for all assessments made for each course related to that outcome that is required of software engineering majors. Table B-5 shows which CSSE courses required of SE majors relate to which SE Program Outcomes from SE1 to SE9. (The CSSE courses listed are discussed in detail in Section B.8.)

In 2005-06, this success criterion holds for all the CARs, with the following exceptions:

- CSSE 220 Course Assessment Report (fall) – 3 outcomes did not hold
- CSSE 230 Course Assessment Report (spring) – 1 outcome did not hold
- CSSE 332 Course Assessment Report (winter) – 1 outcome did not hold
- CSSE 332 Course Assessment Report (spring) – 1 outcome did not hold

This meant that, according to the success criteria, SE1, SE4 and SE5 did not hold. However, those Program Outcomes were successful in most cases:

- SE1 held for 20 of the 22 Course Assessment Reports involving it (90.9%),
- SE4 held for 9 out of the 11 Course Assessment Reports involving it (81.8%), and
- SE5 held for 4 out of the 6 Course Assessment Reports involving it (66.7%).

Table B-5. Map of CSSE courses required of SE majors to SE Program Outcomes.

		Software Engineering Program Outcomes								
		Apply SE	Proto-types	Metrics	Teams	Comm.	Reqs.	Impact	Ethics	Life-long
CSSE #	**Name of CSSE Required Course**	**SE1**	**SE2**	**SE3**	**SE4**	**SE5**	**SE6**	**SE7**	**SE8**	**SE9**
120	Soft. Development I	X			X					
220	Soft. Development II	X								
230	Soft. Development III	X								
232	Comp. Architecture I	X			X	X				
304	Prog. Lang. Concepts	X								
332	Operating Systems	X			X	X				
333	Database Systems	X								
371	Software Req. & Spec.	X	X		X		X			
372	Software Proj. Mgmt.	X		X	X	X		X	X	X
373	Formal Methods	X								
374	Soft. Arch. & Des. I	X			X		X			
375	Soft. Const. & Evol.	X				X				
376	Soft. Quality Assur.	X								
377	Soft. Arch. & Des. II	X								
497-9	Senior Project I-III	X			X	X	X	X	X	

Application Domain Curricular Requirements

The minimal requirements for an application domain track at Rose-Hulman is that 1) it must be related to a significant software application area, and 2) it must contain at least 12 hours worth of courses taught outside of the CSSE department. (The tracks themselves, and the process used for approval of such a track, can be found in Section B.8.)

Since there are no Course Assessment Reports available for any non-CSSE courses used in an application domain, the CARs cannot be used for assessment of SE10 (regarding the application domain) as they are for the other Program Outcomes. However, the CSSE department assumes that those other departments are showing due diligence in their own course assessment.

So, the assessment of the Application Domain Curricular Requirements is twofold:

1. Determination that a software engineering graduate has fulfilled the course requirements for an application domain track.
2. Determination that a software engineering graduate has fulfilled any non-course requirements for an application domain track (e.g. the World Political Studies track requires some type of international experience related to the domain track and approved by the CSSE department head).

This assessment is performed by the Software Engineering Coordinator for inclusion in the Program Assessment Report. The 2005-06 PAR states that all SE graduates for that academic year has fulfilled those requirements (Appendix III.B).

Summary of 2005-06 Program Outcomes Assessment

The Software Engineering Program Assessment Report (Appendix III.B) summarized the assessment results:

- Success was reported from the Exit Interview results for all Program Outcomes.
- Success was reported from assessment of Application Domain Curricular Requirements for all software engineering graduates
- Success was reported from the Course Assessment Reports for all except Program Outcomes SE1, SE4 and SE5. However, despite the unsuccessful cases, the CSSE department is satisfied with the Program Outcomes so far, since even those Outcomes that technically failed were successful in a supermajority of cases. (The bar appears to be too high for the course assessment reports, and it needs to be re-evaluated for next year.)

Evolution of Program Outcomes

As previously stated, the current method for software engineering program assessment (Figure B-2) did not go (fully) into effect until the 2005-06 school year. (However, the course assessment process was tested on several courses during the winter and spring of 2004-05.) Until then, the Program Outcomes were reviewed by the CSSE faculty and Board of Advisors. Table B-6 provides a timeline of review and changes in the Program Outcomes to date.

Table B-7 contains the annual timeline for assessment starting in using the process outlined in Figure B-2.

Table B-6. Timeline for development, review and modification of Program Outcomes to date.

Date	Activities
October 2002	• Input received on potential Program Outcomes by CSSE Board of Advisors
January 2003	• SE Program Outcomes approved by CSSE faculty
May 2003	• SE Program Outcomes reviewed by CSSE Board of Advisors
October 2003	• SE Program Outcomes reviewed by CSSE Board of Advisors
October 2004	• SE Program Outcomes reviewed by CSSE Board of Advisors • SE program outcomes reviewed by CSSE faculty. One change was made: "design and run experiments with software metrics" was changed to "select and use software metrics" in outcome SE3.
Nov 2004-May 2005	• Course assessment process tested on several courses during the winter and spring terms.
August 2005	• SE program outcomes reviewed by CSSE faculty
Sept 2005 – May 2006	• Annual assessment of software engineering program begins • Course Assessment data collected; Course Assessment Reports generated
October 2005	• SE program outcomes reviewed by SE mock accreditation reviewer
June 2006	• The 2005-06 Program Assessment Report, with the results of SE program outcomes assessment, submitted by the Software Engineering Coordinator, with no changes in Program Outcomes recommended.

Table B-7. Timeline for assessment and improvement of Program Outcomes over an academic year.

Time of Year	Activities
August	• CSSE Faculty approve any change in SE Program Outcomes • SE Program Coordinator develops Program Assessment Plan for next year • CSSE faculty approve new SE Program Assessment Plan • Course coordinators and the rest of the course coordination team determined and approved by CSSE faculty • Course Outcomes and Course Assessment Plans for fall courses approved first by Course Coordinators and then by CSSE faculty
Sept-May	• Any software engineering curriculum changes proposed by CSSE faculty and approved by the RHIT curriculum committee
Fall term	• Course Assessment Data for the fall term courses collected • Course Outcomes and Course Assessment Plans for winter courses approved first by Course Coordinators and then by CSSE faculty
Winter term	• Course Assessment Data for the fall term courses collected • Course Assessment Reports for winter courses approved first by Course Coordinators and then by CSSE faculty • Course Outcomes and Course Assessment Plans for spring courses approved first by Course Coordinators and then by CSSE faculty
Spring term	• Course Assessment Data for the fall term courses collected • Course Assessment Reports for spring courses approved first by Course Coordinators and then by CSSE faculty
Summer	• Application Domain Curricular Requirements of 2005-06 graduates assessed by SE Program Coordinator • Course Assessment Reports for spring courses approved first by Course Coordinators and then by CSSE faculty • SE Program Coordinator submits Program Assessment Report to CSSE faculty • CSSE faculty approves Program Assessment Report

c. Achieving ABET Outcomes (a) through (k)

Rose-Hulman has defined ten Institutional Outcomes that all RHIT programs are required to achieve, as shown below. *Note: How the software engineering program achieves the criteria within RH1 through RH6 are listed in italics.*

RH1. Ethics - A recognition of ethical and professional responsibilities

When given the opportunity, students will:

1. Demonstrate knowledge of a professional code of ethics.
   CSSE 372 (Software Project Management) Ethics debate homework assignment (write a summary report of the debate)

2. Evaluate the ethical dimensions of professional engineering, mathematical, and scientific practices.
   CSSE 372 (Software Project Management) Ethics debate homework assignment (write a summary report of the debate)

RH2. Contemporary Issues - An understanding of how contemporary issues shape and are shaped by mathematics, science, & engineering

When applying the principles of mathematics, science, and/or engineering to a technical problem, students will:

1. Demonstrate an awareness of how the problem is affected by social concerns and trends.
   CSSE 371 Vision document and supplemental specification

2. Demonstrate an awareness of how the proposed solution(s) will affect culture and the environment.
   CSSE 371 Vision document and supplemental specification

RH3. Global - An ability to recognize the impact of global societies on citizens and professionals

When given the opportunity, students will:

1. Demonstrate an awareness of the historical development of cultures and societies.
   Provided by GL (Global Studies) courses in the Humanities and Social Sciences department

2. Show an awareness of the relationships of nations and the interdependence of peoples around the globe.
   Provided by GL (Global Studies) courses in the Humanities and Social Sciences department

RH4. Culture - An ability to understand diverse cultural and humanistic traditions

When given the opportunity, students will:

1. Perform, interpret, analyze, or otherwise engage in artistic, literary, and/or other forms of culture.
   Provided by GL (Global Studies) courses in the Humanities and Social Sciences department

2. Interpret, analyze, or evaluate aspects of a society other than the student's own.
   Provided by GL (Global Studies) courses in the Humanities and Social Sciences department

RH5. Teams - An ability to work effectively in teams

When assigned to teams, students will:

1. Share responsibilities and duties.
   CSSE 372 (Software Project Management) Project plan

2. Analyze ideas objectively to discern feasible solutions and then build consensus.
   CSSE 232 (Computer Architecture I) Design process journal

3. Develop a strategy for action.
   CSSE 372 (Software Project Management) Project plan

RH6. Communication - An ability to communicate effectively in oral, written, graphical, and visual forms

When performing communication tasks, students will:

1. Identify readers/audience, assess their previous knowledge & information needs, and organize/design information to meet the needs.
   Provided by RH 330 (Technical Communication)

2. Provide content that is factually correct, supported with evidence, explained with sufficient detail, and properly documented.
   Provided by RH 330 (Technical Communication)

3. Test readers/audience response to determine how well ideas have been relayed.
   Provided by RH 330 (Technical Communication)

4. Submit work with a minimum of errors in spelling, punctuation, grammar, and usage.
   Provided by RH 330 (Technical Communication)

It is also believed that all students should be able to demonstrate the following characteristics with the performance indicators to be decided by each degree granting department:

RH7. Problem Solving - An ability to apply the skills and knowledge necessary for mathematical, scientific, and engineering practices

RH8. Interpreting Data - An ability to interpret graphical, numerical, and textual data

RH9. Experiments - An ability to design and conduct experiments

RH10. Design - An ability to design a product or process to satisfy a client's needs subject to constraints

(How the software engineering program provides these outcomes is discussed later in this section.)

RH1 through RH10 maps to ABET Criterion 3 outcomes (a) through (k) as shown in Table B-8.

Table B-8. ABET a-k outcomes mapped to Rose-Hulman Institutional Outcomes.

ABET Criterion 3 Program Outcomes		**RHIT Institutional Outcomes**									
		Ethics	Contemporary Issues	Global	Culture	Teams	Communication	Problem Solving	Interpreting Data	Experiments	Design
		RH1	**RH2**	**RH3**	**RH4**	**RH5**	**RH6**	**RH7**	**RH8**	**RH9**	**RH10**
Applying math, science & engr.	**a**							✓			
Experiment & interpret data	**b**								✓	✓	
Design	**c**										✓
Teams	**d**					✓					
Problem solving	**e**							✓			
Prof. & ethical responsibility	**f**	✓									
Communications	**g**						✓				
Global society	**h**		✓	✓							
Lifelong learning	**i**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Contemporary issues	**j**		✓								
Techniques, skills, tools	**k**						✓				

Assessment of RH1 through RH6

Data for outcomes RH1 through RH6 are collected and rated through RosE Portfolio, as discussed below.

RosE Portfolio (REPS)

The RosE Portfolio is an online, digital system developed at Rose-Hulman Institute of Technology. It is designed to allow students, faculty, and administrators to archive, assess, and evaluate student work for the purpose of class, department, program, and institutional assessment. In addition, the RosE Portfolio offers features like the Showcase Resume, the Curriculum Map, and customizable Assessment Outlines.

The Commission on the Assessment of Student Outcomes (CASO) began development of the RosE Portfolio in 1996. CASO established two objectives in developing an electronic portfolio for the purpose of student learning outcomes assessment. First, CASO wished to create a data collection method that would allow students to submit documents they produced in technical and non-technical courses (authentic evidence), rather than using standardized tests, documents produced solely for the purposes of assessment, or surveys. Second, CASO wished to develop an online portfolio that would allow for submission, rating, and evaluation of evidence of student learning within a single electronic system. While the RosE Portfolio System (REPS) has been used since 1998, a new REPS software product was developed in the summer of 2004 in cooperation with Quest Information Systems, Inc. The 2005 Rating Session was the first time the new software was used for a full rating session.

REPS consists of the following components:

- Student submission interface: allows students to submit documents in any format to an established set of student learning outcomes
- Faculty interface: allows faculty to verify that students have made the required submissions to the appropriate student learning outcomes; also allows faculty to use the Curriculum Map function, a process in which the faculty member maps his/her course outcomes to Institute-level outcomes
- Rater interface: allows faculty raters to assess student submissions against a defined set of evaluation rubrics. The results of the rating sessions are reported back to all departments on campus and are applied to the further development of the programs.

<u>Purpose of Rating</u>

The purpose of the RosE Portfolio (REPS) Rating Session is to assess evidence of student learning in six non-technical Institute outcomes. These outcomes are grouped into two sets: Communication, Ethics, Teams; and Contemporary Issues, Culture, Global. Each learning outcome is defined by a set of performance criteria (Attach Institute Learning Outcomes).

Evidence of student learning in these six outcomes is collected each year through assignments made by faculty in technical and non-technical departments. For example, some engineering faculty require that students submit documents from capstone senior design courses as evidence for the Teams outcome. Furthermore, Humanities and Social Sciences faculty require that students submit documents produced in their courses for evidence of the Global and Culture outcomes. Although evidence of student learning in all six outcomes is collected every year, the student submissions to the portfolio are rated on a regular cycle (See Table B-9).

Table B-9. RosE Portfolio rating cycle.

Objective	2003-04	2004-05	2005-06	2006-07	2007-08	2008-09
Communications	✓		✓		✓	
Ethics	✓		✓		✓	
Teams	✓		✓		✓	
Contemporary Issues		✓		✓		✓
Culture		✓		✓		✓
Global		✓		✓		✓

Definition, collection of documents, and assessment and evaluation of evidence for technical learning outcomes (both at the Institute and department levels) is the province of technical departments. Some departments elect to use REPS for this data collection and assessment.

Rating Scope and Methodology

Rating submissions to the RosE Portfolio has followed the same methodology since the system was initiated in 1998. Rose-Hulman faculty are hired as portfolio raters. Attempts are made to involve faculty from many different departments on campus to ensure objectivity in rating and broad based familiarity with and participation in the process. Raters work for two days in the Dynamics Laboratory (Olin 203) and are compensated at the rate of $250 per day. The Rating Session Coordinator facilitates the process and assigns pairs of raters to rate student submissions for a particular outcome. For example, a mechanical engineering faculty member and a chemistry faculty member may work as a rating pair assessing the student files under Criterion 4 of the Communication Outcome.

The rating process consists of three steps.

1. First, REPS requires that the faculty rater pair rate a set of three shared documents. The rating is made on the basis of a Rating Rubric and uses a Yes/No benchmark for a single rating question: “Does this submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman.” The raters must agree in their rating of the initial document set. This step is called Inter-Rater Reliability (IRR). The process ensures that raters look for the same qualities and components in order to rate documents as acceptable.

2. Second, if the raters agree in their IRR, the system then allows them to proceed with a set of ten documents, each rater reading and rating a different set of ten. The system records their rating for each document. The system also introduces a shared file every twelve documents in order to check that the raters have maintained their Inter-Rater Reliability. Failure to rate the shared document identically will cause the system to stop the raters so that they can recalibrate their work before moving on to another document set, ensuring validation of the rating process.

3. Third, the raters can provide comments about the rating session or about the student submission in the Comment boxes. In addition to the work of rating, faculty raters are asked to identify Exemplary Documents among the files that they assess.

For the Rating Session, portfolio raters use the Outcome, Performance Criteria, and Rubrics that have been developed by CASO based on feedback from faculty in all departments. The list of faculty raters, their home departments, and their assigned outcome is provided in Table B-10 (from the most recent Rating Session, Summer 2005).

Table B-10. Faculty rater by outcome.

Objective	Criterion	Rater	Department
Ethics	1 and 2	Heinz Luegenbiehl	HSS
	1 and 2	Anneliese Watt*	HSS
Communication	1 and 3	Rebecca DeVasher	Chemistry
	1 and 3	Bill Weiner *	ABBE
	2	Glen Livesay	ABBE
	2	Don Richards*	ME
	4	Bruce Black	ECE
	4	Clark Merkel	ME
	4	Michael Mueller*	Chemistry
	4	Renee Rogge	ABBE
Teams	1	Mark Ardis*	CSSE
	1	Jim Hanson	CE
	1	Sudipa Kirtley*	PHOE
	1	Bill Kline	EM
	2 and 3	David Erwin*	Chemistry
	2 and 3	Richard House	HSS

Faculty who have served as raters in previous Rating Sessions are marked with an "*".

Two key components characterize REPS and distinguish it from other data assessment methods. First, faculty raters are not reading and evaluating documents that were produced in their own courses. Instead faculty rate submissions independent from the courses in which they were produced. This contributes to the validity of the rating process. Second, faculty evaluate documents against a set of defined rubrics (Attach Rating Rubrics).

After the Rating Session results are compiled, they are used by technical departments for preparation of their own ABET reports, and by the academic dean and technical departments for their Higher Learning Commission accreditation reports, as well as reporting to constituencies interested in the evaluation and continuous improvement of RHIT.

Portfolio Usage Comparison

In the 2003 Rating Session, 2,207 unique submissions were made to REPS. During the 2005 Rating Session, 6,928 unique submissions were made to REPS. This represents a significant increase in usage of the system since 2003, the last time that submissions to the Communication, Ethics, and Teams objectives were rated. In 2003, only documents collected during one academic year were rated. In 2005, we rated documents that had been collected for two years, reflecting the Portfolio Rating Cycle (see Table B-9).

Assessment Criteria for RH7 through RH10

As previously stated, Institute Outcomes RH7 through RH10 needs to be demonstrated with performance indicators to be decided by each degree granting department: The CSSE department has developed a "Key Three" method for determining those outcomes using the following criteria:

- Problem Solving Criteria
- Experiments Criteria
- Design Criteria

The definition of each of these criteria are described below.

Problem Solving Criteria

1. Given a specific implementation of a programming problem, analyze the run time of that implementation.

    - Suggested artifact for assessment: CSSE230 (Fundamentals of Software Development III) Exam question

    - Description: Students should be able to establish that they understand how to find the runtime of code containing loops.

    - Example: When asked to find the complexity of a block of code including multiple loops, students should recognize that nested loops increase the complexity of code, while sequential loops do not. As a result, they will give the correct big-oh analysis in each case.

    - Not acceptable: Students give the wrong big-oh analysis for the nested loops or multiply individual loop runtimes when finding the runtime of a series of sequential loops.

2. Design an algorithm to solve a given problem efficiently.

    - Suggested artifact for assessment: CSSE230 (Fundamentals of Software Development III) Exam question

- Description: Students should be able to produce an algorithm that correctly solves a given problem efficiently using the design conventions specified in the course.

- Example: Given two sets implemented as sorted arrays, the student can correctly write an algorithm to find the union of the two sets. Key ideas include incrementing the loop indices correctly to include all elements but avoid duplicates. The runtime should be linear in the size of the arrays.

- Not acceptable: Not enough detail is provided about when to increment the loop indices. All the elements in one array that are greater than the largest element in the other array are ignored because the student terminates the loop improperly. Solution ignores the fact that the arrays are sorted, thus requiring a quadratic algorithm.

3. Given a detailed specification of a programming problem, implement that specification in an appropriate programming language correctly.

    - Suggested artifact for assessment: CSSE220 (Fundamentals of Software Development II) BigRational lab exercise

    - Description: Students should implement the abs, add, divide, multiply, negate, and subtract methods along with appropriate constructors. The implementations should be correct and follow the style conventions specified in the course.

    - Example: See the sample solution to the BigRational project.

    - Not acceptable: One or more of the implementations are seriously incorrect. Style has gross violations of the style conventions specified in the course.

4. Given a proposed implementation of a detailed specification of a programming problem, unit-test that proposed solution in an effective manner.

    - Suggested artifact for assessment: CSSE220 (Fundamentals of Software Development II) BigRational lab exercise

    - Description: Students should provide unit-tests of the abs, add, divide, multiply, negate, and subtract methods. The unit-tests should be provided by using a tool like JUnit. The unit-tests need not be exhaustive but, taken collectively, should demonstrate an understanding of covering a variety of cases including extreme cases.

    - Example: See the sample solution to the BigRational project.

    - Not acceptable: One or more of the methods/constructors is not tested at all. (But note that it is normal for a JUnit test of one method to be an effective test of several methods.) The unit tests fail to include any tests of extreme cases. The unit tests fail to include more than one test per method/constructor.

Experiments Criteria

The following four criteria are used by the Department of Physics for assessing the introductory physics labs:

1. Use appropriate experimental techniques and procedures
2. Keep detailed and accurate records of experimental procedures and data collected
3. Estimate experimental uncertainties and use appropriate error propagation techniques
4. Draw reasonable conclusions from results

Software Engineering (as well as Computer Science) majors are required to take the first two physics courses, PH 111 and 112 (Physics I and II). The CSSE department receives assessment data from the Physics department on their introductory labs on an annual basis.

Design Criteria

When given the opportunity, students will:

1. Develop a design specification that addresses customer/client needs and constraints.
    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Requirements Specification
    - Description: The student should describe all of the customer's requirements for the product or process.
    - Example: CSSE 497-498-499 (Senior Projects I-III) Requirements Specification including use cases that describe the flow of activities for the product or process.
    - Not acceptable: insufficient coverage of use cases
2. Carry out a conceptual design by generating multiple solutions that address the issues above, evaluating the feasibility of the solutions, and choosing the appropriate solution.
    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Architecture Document
    - Description: The student should create a high-level design whose features and capacities are traceable to the key requirements described in the problem statement and requirements document.
    - Example: CSSE 497-498-499 (Senior Projects I-III) Architecture Document
    - Not acceptable: The architecture document does not describe how key required features and capacities will be achieved by that design.

3. Carry out a detail-level design and implementation using appropriate design tools and methodologies**.**

    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Software Deliverables
    - Description: The student should detail their design in a manner that meets strategic goals of the project, such as implementing required features in a manner which easily can be maintained by others.
    - Example: CSSE 497-498-499 (Senior Projects I-III) Software Deliverables
    - Not acceptable: The implementation does not follow design guidelines expressed in the architecture document. For instance, a design template for error handling, required by the architecture, is not adhered to.

4. Test and refine the implementation until the product or process design specifications are met or exceeded.

    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Test Results Report
    - Description: The student should describe the planned test cases and the results of running the tests.
    - Example: CSSE 497-498-499 (Senior Projects I-III) Test Results Report
    - Not acceptable: Insufficient testing planned or executed.

5. Document the finished product or process as appropriate for the discipline according to standard practice.

    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) User Documentation
    - Description: The student should provide instructions in the use and maintenance of the product or process.
    - Example: CSSE 497-498-499 (Senior Projects I-III) User Documentation
    - Not acceptable: Insufficient documentation provided

6. Present and transfer the product or process and documentation to the client.

    - Suggested artifact for assessment: CSSE 497-498-499 (Senior Projects I-III) Project presentation at expo
    - Description: The student should prepare a description of the finished product or process for display at the Senior Project Expo
    - Example: CSSE 497-498-499 (Senior Projects I-III) Expo Poster
    - Not acceptable: Poorly-designed or incoherent description of project results.

Evaluation and Assessment Results to Date

By the end of the summer, the SE program will for the first time have a complete set of assessment data for all of the Institute Outcomes (RH1 through RH10). These results will then be discussed at the annual CSSE faculty retreat on August 22, 2006, and appropriate actions taken if necessary. (The results will be made available to the software engineering program evaluator during the site visit.)

The data collected to date is discussed below.

Assessment of RH1 through RH6

The first software engineering graduates were during the 2004-05 school year. As stated in Table B-9, the Contemporary Issues, Culture and Global outcomes were evaluated in the summer of 2005. The results for software engineering majors from that evaluation is shown in Table B-11.

The Communications, Ethics and Teams outcomes will be evaluated sometime in the summer of 2006, with results expected in time for the fall retreat.

Assessment of RH7 through RH10

Problem Solving Criteria:

- For Criteria 1, 80.8% of submissions in CSSE 230 (Fundamentals of Software Development III) during spring 2006 term met the criteria.
- Criteria 2-4: results not yet available.

Experiments Criteria:

The Physics department conducted the assessment by sampling the RC Circuits reports submitted for the PH 112 (Physics II) Lab during the winter 2005-06 term. Each lab instructor was asked to collect certain student labs which were then copied and returned to the students. The actual assessment was conducted during the spring term by members of the Physics Curriculum Committee.

Three committee members discussed each criteria and came up with guidelines on what would be acceptable and unacceptable student performance. Five reports where evaluated by each assessor and the results discussed. Finally, each assessor evaluated a set of reports independently. Any time there was any question about how a report should be rated, it was discussed as a group.

The results are shown in Table B-12.

Table B-11. Summer 2005 RoseE Portfolio rating results for software engineering majors.

RHIT Learning Objectives	Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (#)							
	FR		SO		JR		SR	
A. Ethics	Yes	No	Yes	No	Yes	No	Yes	No
Criterion 1			1		1		1	
Criterion 2			1		7		2	
	FR		SO		JR		SR	
E. Teams	Yes	No	Yes	No	Yes	No	Yes	No
Criterion 1	7	8	5	5				
Criterion 2	1					1		1
Criterion 3	2				1		1	
	FR		SO		JR		SR	
F. Communication	Yes	No	Yes	No	Yes	No	Yes	No
Criterion 1		1		1	4		3	
Criterion 2			2	1	3	1	2	1
Criterion 3			1		2	2		3
Criterion 4	9	4	5	3	1	3	2	1
Total	19	13	15	10	19	7	11	6

RHIT Learning Objectives	Does the submission meet the criterion at a level expected of a student who will graduate from Rose-Hulman? (%)							
	FR		SO		JR		SR	
A. Ethics	Yes	No	Yes	No	Yes	No	Yes	No
Criterion 1			100%	0%	100%	0%	100%	0%
Criterion 2			100%	0%	100%	0%	100%	0%
	FR		SO		JR		SR	
E. Teams	Yes	No	Yes	No	Yes	No	Yes	No
Criterion 1	47%	53%	50%	50%				
Criterion 2	100%	0%			0%	100%	0%	100%
Criterion 3	100%	0%			100%	0%	100%	0%
	FR		SO		JR		SR	
F. Communication	Yes	No	Yes	No	Yes	No	Yes	No
Criterion 1	0%	100%	0%	100%	100%	0%	100%	0%
Criterion 2			67%	33%	75%	25%	67%	33%
Criterion 3			100%	0%	50%	50%	0%	100%
Criterion 4	69%	31%	63%	38%	25%	75%	67%	33%
Total	59%	41%	60%	40%	73%	27%	65%	35%

Table B-12. Results of assessment of experiments criteria in PH 112 labs.

Criteria	Acceptable	Not Acceptable
1	28	11
2	33	6
3	22	17
4	15	24

Design Criteria

The results from assessment of the senior projects teams from 2005-06 can be found in Table B-13.

Table B-13. Results of Design Criteria assessment for 2005-06 senior project teams.

Team	Criterion 1: Reqs.	Criterion 2: Architecture(9)	Criterion 3: Impl.	Criterion 4: Test	Criterion 5: Doc.	Criterion 6: Expo
1	Yes	Yes	Yes	No(1)	Yes	Yes
2	Yes	Yes	Yes	Yes	Yes	Yes
3	Yes	Yes(2)	Yes(2)	No(1)	No(3)	Yes
4	Yes	Yes(4)	Yes(4)	Yes(5)	Yes(6)	Yes
5	Yes	Yes	Yes	Yes (11)	Yes	Yes
6	No (7)	Yes	Yes	Yes	Yes	Yes
7	Yes	No (10)	Yes	No (10)	No (10)	Yes
8	No (8)	Yes	Yes	No (8)	No (8)	Yes
9	Yes	Yes	Yes	Yes	Yes	Yes
TOTAL	7 of 9	8 of 9	9 of 9	5 of 9	6 of 9	9 of 9
Notes						
1	No test plan or test cases.					
2	Extensive comments in code describe design.					
3	But client did not require this.					
4	Used format specified by client for these.					
5	Team interacted verbally with client several times a week about test cases.					
6	Handoff was to client's developers, with documentation in code.					
7	Did not prepare a written requirements specification.					
8	Did not turn in evidence of this activity.					
9	In some cases project plans were used to determine this category.					
10	The nature of the project (exploratory) did not allow for this.					
11	Most comprehensive test planning and execution I have ever seen!					

4. Professional Component

Students are prepared for practice as a software engineer through courses in software engineering, computer science, computer engineering, continuous and discrete mathematics, laboratory science, humanities and social sciences, and an software application domain track. A breakdown of topics by area and number of credit hours is shown in Table B-14.

Table B-14. The software engineering program by topic and credit hours.

Subject	**# Credit Hours**
Software Engineering - required	40
Computer Science - required	24
Restricted CSSE Elective	4
Mathematics - required	31
Math Elective	4
Computer Engineering	8
Lab Sciences	16
Humanities and Social Sciences	36
Free Electives	28
College and Life Skills (for incoming freshmen)	1
Total Number of Credit Hours	**192**
Application Domain Track (The application domain track courses can be used to count in any of the elective subject areas, depending on the course subject)	12-22

As far as the required areas of the professional component are concerned:

- Engineering topics are fulfilled through 76 hours (39.6% of the 192 total hours in the SE curriculum) of software engineering, computer engineering and computer science (as engineering science) courses
- Mathematics and Basic Sciences are satisfied through 51 hours (26.6%) of mathematics and lab science courses, and
- General education consists of 37 hours (19.3%) satisfied through 36 hours of humanities and social science (HSS) plus a one-hour freshman class in College and Life Skills.

Table B-15 shows the breakdown of courses by the required areas of the professional component.

All students admitted to Rose-Hulman, regardless of major, are assumed to have sufficient math background to take calculus and enough science background to take college introductory courses in physics, chemistry and biology. The student's high school math and science background plus the mathematics and lab science courses required in the software engineering curriculum collectively serve to provide the appropriate fundamentals in these areas that are needed for the

engineering topics. The engineering courses culminate in a capstone project with a major design component.

Table B-15. Breakdown of required areas of the professional component.

Subject	Number of Credit Hours	ABET/EAC Requirement	% of Total (192 hours)
Engineering Topics	76	72	39.6%
Mathematics and Basic Sciences	51	48	26.6%
General Education	37	Sufficient to complement technical component	19.3%

The engineering topics are distributed as follows:

- 72 credit hours of CSSE courses, including
  - 12 hours of introductory courses with significant content in both CS and SE
  - 40 hours of required software engineering courses
  - 16 hours of required computer science courses
  - 4 hours in a restricted CSSE elective
- 4 credit hours in computer engineering

The particular courses used for the engineering topics are described in Section B.8.

The courses satisfying the Mathematics and Basic Sciences portion of the Professional Component are described in Table B-16

Table B-16. Courses satisfying the ABET Mathematics & Basic Science requirement.

Subject and #	Course Name	Hours Towards Professional Component
MA 111	Calculus I	5
MA 112	Calculus II	5
MA 113	Calculus III	5
MA 221	Differential Equations and Matrix Algebra I	4
MA 275	Discrete Combinatorial Analysis I	4
MA 375	Discrete Combinatorial Analysis II	4
MA 381	Intro to Probability with Statistical Applications	4
PH 111	Physics I	4
PH 112	Physics II	4
CHEM 201	Engineering Chemistry I	4
	Mathematics (MA) elective	4
	Science elective (in PH, CHEM or Applied Biology)	4
Total		**51**

The general education component consists of the 36-hour Institute requirement for Humanities and Social Sciences (HSS) as well as a one hour of CLSK 100 (College and Life Skills) course required of incoming freshmen. The mission and objectives of the HSS Department are published on the department homepage http://www.rose-hulman.edu/hss/department.htm and in the official Course Catalogue.

The HSS Department consists of 24 full-time faculty teaching courses in 11 disciplines. Over 150 courses are offered by the department each academic year. Every Rose-Hulman student must take a total of 36 credit hours in the HSS Department in order to fulfill the Institute graduation requirements. As part of those 36 hours, all students are required to take RH 131 Rhetoric and Composition (a freshman-level composition course). In addition, engineering programs may specify additional courses appropriate to their majors; software engineering programs at Rose-Hulman require that students take RH 330 (Technical Communication) as one of their 36 credit hours.

Although each faculty member in the department is a specialist in a particular academic discipline, the organizational principle of HSS course offerings is interdisciplinary; all HSS courses are distributed, therefore, into 4 thematic categories. In order to provide students with a broad education in the non-technical disciplines, each student is required to take 8 credit hours in each of the 4 thematic categories (32 credit hours):

- **Global Studies (GL prefix):** Courses whose primary focus is on non-Western cultural traditions, or on the interrelationships among multiple societies.
- **Rhetoric and Expression (RH prefix):** Courses whose primary focus is on human communication in all its forms.
- **Self and Society (SL prefix):** Courses whose primary focus is on the dynamics and patterns of human and social interaction.
- **Values and Contemporary Issues (VA prefix):** Courses whose primary focus is on societal values and/or on current topics of importance.

Thus, a student who chooses to take GL 184 Introduction to East Asia (taught by an anthropology professor) and GL 261 Comparative Politics (taught by a political science professor) would satisfy the HSS requirement of taking 8 credit hours in the Global Studies category. To complete the 36 credit hour requirement, a student takes the remaining 4 credit hours with the required freshman-level writing course.

5. Faculty

Rose-Hulman is fortunate to have high-quality faculty in sufficient numbers to cover all CSSE courses – including those in software engineering – with sections of maximum size 30 in virtually all cases. In the sections below, the size of the faculty and course sections, faculty-student interaction, and the faculty member qualifications are discussed.

a. Faculty and Course Section Size

There are currently 12 CSSE faculty (all full-time, tenure-track with PhDs), with three primarily teaching software engineering and other nine computer science. There are normally four SE faculty members; it is expected that another software engineering faculty member will be hired starting in the 2007-08 school year, to fill a recent vacancy. Due to sabbaticals, eleven CSSE faculty are normally on-campus during a particular academic year (and in fact, the departing SE faculty member was originally scheduled to be on sabbatical in 2006-07).

The standard course load for all CSSE faculty is three sections per term for each of the three ten-week terms, with no more than two different course preparations per term. Two faculty members have administrative release time: Dr. Cary Laxer (CSSE department head) teaches only one course per term and Dr. Donald Bagert (Director of Software Engineering) teaches two. That means that there are 90 faculty-sections per year available for instruction of CSSE courses. This means that there are more than sufficient CSSE faculty to teach all required courses (both lecture and labs) and a wide variety of electives and teach them in sections with no more than 30 students each. It is also not uncommon for faculty to get release time to prepare a potential new course for the next term and to then subsequently teach it as a special topics class to see if it should be converted into a permanent course.

Table B-17 shows the maximum number of sections needed for each of the ten required software engineering courses in a year. (Some of these courses are required of – or in the case of senior projects – usually taken by computer science majors. CS majors also sometimes take CSSE 373, 374, 375 and 376 as free electives, with the latter course – Software Quality Assurance – as the most frequently taken.)

Two of the three current SE faculty (Dr. Steve Chenoweth and Dr. Curt Clifton) have the standard load, while Dr. Donald Bagert, as Director of Software Engineering, has a reduced load of two sections per term. So, as a group the current software engineering faculty teaches 24 sections per year, more than the 21 sections needed to cover the SE courses according to Table B-17. However, in order to avoid having one or more faculty members do three preparations in one or more terms in 2006-07, Dr. David Mutchler of the CS faculty will be teaching CSSE 371 and 372 in the fall and winter terms, respectively. (Dr. Mutchler has in past years frequently taught courses in the CSSE introductory sequence, which contains about 40% software engineering content.)

Table B-17. Number of sections needed for each software engineering course each year.

Course	Max # of Sections Needed
CSSE 371 Software Requirements and Specification	3
CSSE 372 Software Project Management	3
CSSE 373 Formal Methods in Specification and Design	1
CSSE 374 Software Architecture and Design I	1
CSSE 375 Software Construction and Evolution	1
CSSE 376 Software Quality Assurance	2
CSSE 377 Software Architecture and Design II	1
CSSE 497 Senior Project I	3
CSSE 498 Senior Project II	3
CSSE 499 Senior Project III	3
Total	**21**

As far as computer science courses are concerned, at least two sections of each of the CS courses required of software engineering majors (CSSE 120, 220, 230, 232, 304, 332 and 333) are taught each year. In addition, there about 15 courses taught each year that can be taken as restricted CSSE electives by software engineering majors, such as computer networks, computer security, two courses in computer graphics, artificial intelligence, computer vision, compiler design and web-based information systems. Once again, virtually every section has no more than 30 students in it, with *all* lectures and labs taught by tenure-track faculty.

It is for these reasons that the CSSE department can firmly state that there are more than sufficient faculty to teach SE majors in all subject areas.

b. Faculty-Student and Professional Interactions

Rose-Hulman faculty members pride themselves as being "hands-on" in their interaction with students, and the CSSE department is no exception. All faculty have an "open door" policy, and spend much of their day helping and advising students in their offices or one of the labs. (Student assistants for courses are used only as graders, to assist the faculty in laboratories, or to have office hours in order to assist students in the evenings or on weekends.) Also, it is not uncommon for faculty to provide students their home number, cell phone number, or instant messenger screen name, and tell those students to contact them if they need help after hours. In the CSSE department, faculty encourage students to use their first names when addressing them. Overall, students feel that the CSSE faculty are not only their mentors, but their friends as well.

Since virtually all CSSE sections are no larger than 30 students, a particular faculty member teaches at most 90 students in a particular term (though the number is usually somewhat smaller than that). Also, previously stated, ratio of students to permanent academic advisors in CSSE is about 16:1 on average, and those acting as freshman advisors have only about 15 students. This means that Rose-Hulman faculty have more than sufficient time to spend teaching, advising and otherwise assisting students.

CSSE faculty are involved in a variety of Rose-Hulman service functions. CSSE faculty serve as advisors for student chapters of the Rose-Hulman Technology Club and the Upsilon Pi Epsilon (UPE) computer science honorary. CSSE faculty are also advisors or are heavily involved in a

number of service groups, fraternities, and religious organizations on campus. Also, each CSSE faculty member usually serves on at least one Institute-wide committee, as well as departmental committees.

Professionally, CSSE faculty attend conferences, make meeting presentations, serve on professional organization and conference committees, publish in professional journals, serve on editorial boards and university boards of advisors, and do research, often with undergraduates. Each faculty member is supported to attend at least one professional meeting per year. In 2005-06 the CSSE faculty attended 30 different conferences and workshops and authored 31 different papers. At Rose-Hulman, virtually no courses are taught during the summer, freeing up the faculty to do research and travel domestically and overseas to professional meetings.

Finally, Rose-Hulman has a strong relationship with industry throughout the United States. Almost all senior projects are done for "real-world" clients; in recent years, this has included Microsoft, Motorola, Beckman Coulter, the Library of Congress, a University of Michigan research group, and a variety of non-profit organizations. Many classes have at least one industry professional come in to guest lecture. Also, the CSSE department has a Board of Advisors which includes members from companies such as Microsoft, Lilly Pharmaceutical, Beckman Coulter, Motorola and Rockwell Collins. The board meets as a group with the faculty at least once a year and is an excellent source of information and advice throughout the year.

c. Faculty Member Qualifications and Experience

One of the finest and most exceptional qualities of the CSSE department is the excellent relationship among all faculty in the department, whether primarily teaching SE or CS. Everyone is focused on the goal of excellence in both computer science and software engineering programs. It is indeed a unique, enjoyable and productive experience to teach in the CSSE department.

The 12 current CSSE faculty are all highly qualified to teach undergraduate computer science and/or software engineering. In this section, the qualifications of the three software engineering faculty are presented in detail, followed by shorter descriptions of the computer science faculty.

Software Engineering Faculty

All three current software engineering faculty members have PhDs in a computing field. One has primarily an academic background, one primarily industrial, and the other a mixture of the two. All have extensive teaching experience, in academia and/or industry.

Dr. Donald Bagert has been a college faculty member continuously for the past 27 years. Originally focusing on computer science, Dr. Bagert first started teaching software engineering courses in 1988. In the late 1990's, he and Dr. Susan Mengel transformed both undergraduate and graduate software engineering at Texas Tech University, creating a Master of Science in Software Engineering, and developing a web-based project process which was used throughout the 12 undergraduate and graduate software engineering courses. By the time he left Texas Tech in 2002, Dr. Bagert had taught a majority of those courses, including classes in software specification and design, software project management, software construction and evolution, software process improvement, and both undergraduate and graduate capstone project sequences.

At Texas Tech, Dr. Bagert also supervised three PhD dissertations and 19 Master's Theses, was for 10 years the principal undergraduate advisor for Computer Science (for as many as 450 students in a year), was Associate Chair of Computer Science for two years (one of these when the Department Chair was primarily working at NASA Ames in California), and created and was co-director of a software engineering research center.

As the Director of Software Engineering at Rose-Hulman, Dr. Bagert led the creation of the Bachelor's degree program in software engineering in his first year there (2002-03). Since his arrival in 2002, he has published over 20 articles (out of more than 120 for his entire career).

Over the years, Dr. Bagert has provided a critical leadership role in the areas of software engineering education, software engineering professional licensing and software professional certification. From 2000-2005 Dr. Bagert was the Chair of the Steering Committee for the IEEE-CS Conference on Software Engineering Education and Training (CSEE&T), the leading conference in its field, and still remains on the committee as Past Chair. Besides CSEE&T, Dr. Bagert has had leadership or organizational roles in the development of three software engineering education and training entities: the Working Group for Software Engineering Education and Training (WGSEET), the IEEE-CS Software Engineering Education Community (SEECo), the Forum for Advancing Software Engineering Education (FASE) electronic newsletter, and the Software Engineering Program Leaders Association (SEPLA).

In 1997-99, Dr. Bagert was one a group of six people (of a total WGSEET committee of 17) which co-authored the *Guidelines for Software Engineering Education*, which was released as a Software Engineering Institute (SEI) technical report and rapidly became the *de facto* worldwide standard for undergraduate SE curricula for the next several years. In 2002, Dr. Bagert was one of the first ABET software engineering program evaluators; in March 2006 he was named to the ABET Engineering Accreditation Commission for 2006-07.

Dr. Bagert has also been involved with the licensing of software engineers in the USA from the beginning. In, 1997-98, he was part of Software Engineering Advisory Committee of the Texas Board of Professional Engineers, which helped the Board develop guidelines for licensing Professional Engineers (PEs) in software engineering in Texas, the first U.S. state to do so. In 1999-2001, Dr. Bagert led the development of the first examination suite for the IEEE-CS Certified Software Development Professional (CSDP) program, and later became the first chair of the CSDP Certification Committee. He is now working with the IEEE Computer Society on a possible certification exam based on SEEK for recent software engineering graduates.

In 1998, Dr. Bagert became the first Professional Engineer licensed in software engineering in both Texas and the United States, and is still has an active PE license in Texas. He is also a Senior Member of IEEE, and an IEEE Golden Core Member (bestowed for professional service).

Dr. Stephen Chenoweth has an educational background which includes a Ph.D. in Computer Science and Engineering as well as an MBA from Wright State University in Dayton. He moved to Dayton, Ohio in 1974 to work for computer vendor NCR; his work included architecting new kinds of transaction processing systems and transferring technologies from research. In 1995 Steve joined Bell Laboratories to serve as an inside consultant, reviewing the architectures of next-generation telecommunications equipment; also among his duties was to teach core software engineering courses to almost 800 Lucent Technologies employees. His research

interests include software architecture and requirements, software processes, artificial intelligence, and alternatives to pedagogy. Dr. Chenoweth joined the CSSE faculty in 2003.

Dr. Curt Clifton is an assistant professor in the CSSE department. Dr. Clifton earned his Ph.D. in Computer Science from Iowa State University in 2005. His research interests include language design and implementation and software engineering, with an emphasis on software modularity issues. He is a co-designer and the main implementer of the MultiJava programming language. He has also worked on the Java Modeling Language for behavior and interface specification and on foundations of aspect-oriented programming. Dr. Clifton also holds a B.S. of Electrical Engineering and a M.S. in Computer Science, both from Iowa State University. Between his undergraduate and graduate study, Dr. Clifton spent six years working in industry as a process engineer for Procter & Gamble, Inc. and as a senior electrical engineer and software team leader for L&S Electric, Inc., a systems integrator for the hydroelectric industry. Dr. Clifton is a member of the IEEE, the IEEE Computer Society, the ACM, SIGSOFT, SIGPLAN, SIGCSE, and ASEE.

All three faculty members have consistently performed between 3 (Satisfactory) and 5 (Excellent) on a five-point Likert scale in overall student evaluations in their courses. In addition, in 2004-05 the SE faculty decided to institute peer evaluations of each other on a rotating basis, in order to be able to better provide input for further course and lecture improvement.

Overall, the software engineering faculty have worked well together in all facets of the SE program, with a high degree of collegiality and enthusiasm by each member. Each summer, the SE faculty have a one to two-day retreat to discuss results of the previous year, and to plan and prioritize for the coming year. (This is in addition to a fall retreat of the entire CSSE faculty.) The SE faculty also meets as needed during the school year to report and discuss the progress of ongoing tasks. (Once again, this is in addition to the weekly CSSE department faculty meeting.)

Computer Science Faculty

The Computer Science faculty are equally qualified to teach their respective courses. All have PhDs, and each one has at least one degree in computer science or computer engineering. A brief overview of educational and research backgrounds of the nine computer science faculty (two currently on sabbatical) are shown below.

- Claude Anderson (Professor, at RHIT since 1988), Ph.D. Illinois (1981, Mathematics) - programming languages, compilers, object oriented programming, theoretical computer science, integration of computer science with other disciplines. (On sabbatical in 2005-06)

- Matthew Boutell (Assistant Professor, at RHIT since 2005), Ph.D. University of Rochester (2005, Computer Science) - computer vision, machine learning.

- Archana Chidanandan (Assistant Professor, at RHIT since 2003), Ph.D. Louisiana at Lafayette (2004, Computer Engineering) - hardware design, image processing, wireless communications.

- Lisa Kaczmarczyk (Assistant Professor, at RHIT since 2005), Ph.D., University of Texas at Austin (2005, interdisciplinary with a focus on Cognitive Science) - cognitive science.

- Cary Laxer (Professor and Head, at RHIT since 1981), Ph.D. Duke (1980, Biomedical Engineering) - computer graphics, data structures, biomedical computing.

- J.P. Mellor (Associate Professor, at RHIT since 1999), Ph.D. MIT (2000, Electrical Engineering, Computer Science) - computer vision, human-computer interaction. (On sabbatical in 2005-06)

- Larry Merkle (Assistant Professor, at RHIT since 2002), Ph.D. Air Force Institute of Technology (1996, Computer Engineering) - data structures, design and analysis of algorithms, artificial intelligence, evolutionary computation, high-performance computing, scientific computation.

- David Mutchler (Professor, at RHIT since 1994), Ph.D. Duke (1986, Computer Science) - artificial intelligence, machine game-playing, probabilistic analysis of algorithms, replicated databases, statistical testing of software.

- Michael Wollowski (Associate Professor, at RHIT since 1999), Ph.D. Indiana (1998, Computer Science) - diagrammatic logic, the World Wide Web, artificial intelligence, cognitive science.

Summary

It is the opinion of the CSSE department that all CS and SE faculty are qualified to teach their respective courses in the software engineering program, that the faculty are professionally active, and that the work environment for both faculty and students is excellent.

6. Facilities

a. Institute-Wide Computer Facilities

Computers and computing are central to the Rose-Hulman experience. Since 1995 all entering students have been required to purchase a laptop computer. The Dell Precision M70 was selected for the 2005-06 school year; it comes with a professional software package that includes Microsoft Office Professional XP, and both wired and wireless network cards. New faculty members receive a freshman laptop for that year, and each faculty member gets a new laptop every three years.

The Technical Services Center manages the main campus computing and networking services. Most classrooms are equipped with desktop power and network connections. Several public areas have Intel-based machines and Sun workstations.

Residence halls are wired for network access. Computers in all classrooms, laboratories and faculty offices are connected to the campus network. All academic buildings, the Student Union and the primary junior/senior residence hall all also provide wireless networking. The campus network has more than seven terabytes of storage.

b. Departmental Computer Facilities

The CSSE departmental computer network is a subnet of the Rose-Hulman network. The department has root access to all of its machines.

The department maintains a number of publicly available Sun workstations including 2 Ultra80s, 12 Ultra 10s, and 12 SunBlade 150s. There are an additional 12 Intel PC desktop computers, most of which are configured for dual boot (Windows and Linux). There are additional machines that provide network and file services for the department's computers. All faculty who want them have a Sun Ultra 5 or Ultra 10 in their offices, in addition to their laptop. Students help manage the entire departmental network as part of their non-academic education.

The department has two general purpose computing laboratories located in the center of the CSSE department office complex. The laboratories are configured so students can meet in small groups around one or two workstations and still have table space for papers, books, and their laptops. A third lab is dedicated for use in the Operating Systems and Computer Security courses. This lab has 12 computers that can be isolated from the department and campus networks for student projects.

In addition, the Imaging Systems Laboratory (an interdisciplinary laboratory jointly operated by the Departments of Computer Science and Software Engineering, Electrical and Computer Engineering, Mathematics, and Physics and Optical Engineering) contains six SGI workstations and two Pentium-based workstations.

c. Software

In addition to Microsoft Office Professional XP, the department provides a variety of software development tools, primarily through the Microsoft Software Development Network Academic

Alliance (MSDNAA). The department's MSDNAA license includes Visio, Visual Studio .NET, Microsoft Project, and Microsoft Source Safe. The primary Java development environment is Eclipse, and Subversion is the primary software used for configuration management.

Among the software tools used in courses required of software engineering majors are

- Eclipse SDK for Java software development, including the JUnit testing environment,
- The Subversion and CVS configuration control systems,
- Microsoft Project for project planning,
- The FPcounter and fpwiz tools for function point estimation,
- The COSMOS tool for cost software cost estimation and
- The ZTC, an environment for the Z formal specification language.

In 2004, Rose-Hulman adopted the ANGEL Course Management System as its course management software. Faculty are encouraged to use ANGEL in their courses.

d. Relationship to Program Educational Objectives

As previous stated, the software engineering Program Educational Objectives are that Software engineering graduates will

PEO1. develop complex systems (including analysis, design, construction, maintenance, quality assurance and project management) using the appropriate theory, principles, tools and processes.

PEO2. use appropriate computer science and mathematics principles in the development of software systems.

PEO3. solve problems in a team environment through effective use of written and oral communication skills.

PEO4. have knowledge of current issues presently involved in effectively performing duties as a software practitioner in an ethical and professional manner for the benefit of society.

PEO5. practice the lifelong learning needed in order to keep current as new issues emerge.

PEO6. develop software in at least one application domain.

The facilities primarily address Program Educational Objectives PEO1 and PEO2. The tools to educate students in the engineering of complex software systems and in computer science fundamentals are available and readily accessible through the three CSSE laboratories and the student laptops.

7. Institutional Support and Financial Resources

All academic departments at Rose-Hulman are organized in a single administrative structure. Consequently there is not a "college of engineering" that must compete with other colleges within a university structure. Because 85% of our students major in engineering and 8% in computer science, it is clear that adequate support of engineering programs is inextricably tied to the continued vigor of the institution. Individual departments will report on program-specific resources.

a. Institute Finances and Faculty Support

Rose-Hulman is in sound financial condition as is indicated by an "A1" rating from Moodys Investors Service. The Institute's operating budget is approximately $70 million per year. Approximately 65% of revenue comes from student tuition. The discount rate is 31%. The Institute currently draws $13,885,000 yearly from is endowment of approximately $160 million.

The Institute's previous 10-year capital campaign which ended in 2004 resulted in $250 million. A new campaign is expected to be launched within two years. Compensation goals described in our 2000 ABET report (faculty salaries at the 60th percentile of the Association of Independent Technological Universities (AITU), professional salaries at the 60th percentile of institutions of similar size as reported by College and University Personnel Association (CUPA), and average hourly staff wages at the 80th percentile of local markets) have become dated. Consequently the Board of Trustees has recently formed a Commission for comprehensive review of the Institute salary structures.

The Institute maintains professional development funds adequate to allow each faculty member to attend at least one national meeting per year. One-third of this support comes from the dean of faculty's budget. The Institute maintains an active sabbatical program that provides one full academic-year release at half pay (or 2 quarters release at 4/5 pay, or one quarter at full pay) every seven years.

Rose-Hulman Institute begins its **budget cycle** early in the academic year. In the fall, all units begin thinking about their budget needs. Each academic department submits budget requests to the Vice President for Academic Affairs. The Vice President for Academic Affairs meets with each department head and may have discussions with the department heads as a whole to help prioritize expenditures and to find possible partnerships which create not only a better use of resources but also a better shared experience.

Typically, first requests for the budget from academic departments are solicited by the Vice President for Academic Affairs (VPAA) in November. After suggesting modifications, the VPAA communicates all these to the Vice President and Chief Administrative Officer (CAO). The CAO prepares a first draft of the budget for presentation to the President. Then a meeting is called that includes all the vice presidents, department heads, and other key section heads. Through a series of these meetings and further budget revisions, a final budget is crafted. The process is transparent and collaborative, which helps us to express our needs, discuss competing interests, and eventually meet our budget goals. The President submits the final budget to the

Executive Committee of the Board of Trustees in late January. They discuss with the President any questions about the proposed budget. The Vice President for Academic Affairs is present in these meetings. Once the Executive Committee has made any suggestions, the President and Vice Presidents consider these suggestions and make any necessary changes. The finalized budget is then submitted to the Board of Trustees at its annual winter meeting, held in late February. The budget includes all elements, ranging from salary to equipment. Once the budget is approved at the February Board Meeting, letters of appointment for the following year are issued to current faculty members.

In the next three sub-sections we describe three major institutional-level developments that have had a major impact during the six years since the previous ABET visit. The three developments in instructional equipment, computer infrastructure, and project space indicate the institutional commitment to constant improvement of our engineering programs.

b. Instructional Equipment

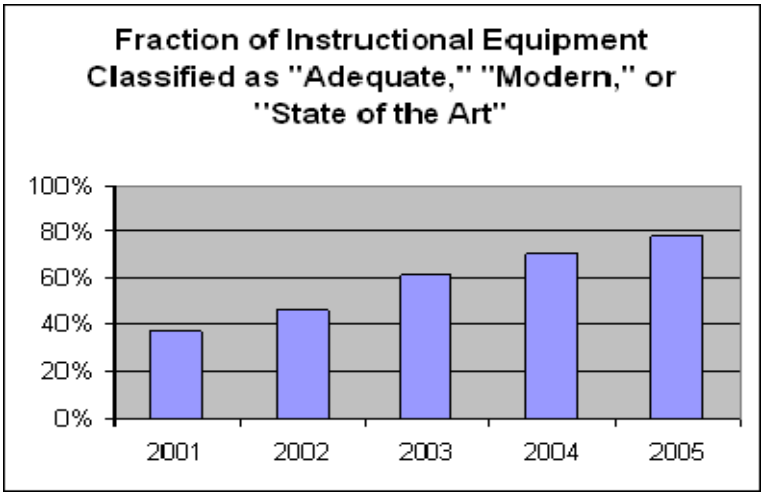
In the summer of 2001 the dean of faculty requested all departments to perform a comprehensive inventory of instructional equipment. The inventory included date of purchase, replacement costs, anticipated lifetime of the item if replaced, and an evaluation of state of current terms, rating each as "state-of-the-art," "modern," "adequate," "obsolete," or "embarrassing."

The survey indicated two things: (1) only 37% of the instructional equipment was rated as state-of-the-art, modern or adequate and (2) that the Institute was under-budgeting instructional equipment by over $1 million per year just to maintain steady state.

As a result, the Institute leadership committed to a strategic goal of increasing the yearly operating budget for instructional equipment by $200,000 per year for each of 5 years in order to eliminate the $1 million per year short fall. As a consequence, the FY 2006-07 budget includes $800,000 more for instructional equipment than did the FY 2002-03 budget.

The summer instructional equipment survey is now a permanent part of our equipment replacement process. Figure B-3 shows the systematic improvement in instructional equipment at the Institute. The summer 2005 inventory indicated that nearly 78% of the Institute's instructional equipment is rated by departments as state-of-the-art, modern, or adequate.

Figure B-3. Improvement in the quality of instructional equipment at Rose-Hulman since commitment to increase reach $1,000,000 yearly allocation.



It should be noted that the increased funds are institutionally budgeted to the dean of faculty, not individual departments. Individual departments make requests each year. These requests are compiled by the dean and distributed to department heads. The dean and assembled department heads then meet to discuss one another's respective needs. In a collegial and Institute-oriented discussion, department heads formulate by consensus their recommendation to the dean for the yearly disbursal of funds.

c. Computer Infrastructure

Rose-Hulman's Information Technology Fabric has been completely transformed since the last ABET visit. Dr. Louis Turcotte began on July 1, 2000 as the Vice President for Instructional, Administrative, and Information Technology. Changes since that date have been truly transformational. The Institute IT system changed from an idiosyncratic collection of early technologies to a true, enterprise computing system. As indication of institutional commitment to educational excellence, the Board of Trustees in 2001 provided a $700,000 budget advance to completely replace our network.

Over a period of approximately nine months the campus networking infrastructure was totally overhauled. Three large router/switches are the primary components and are configured into a triangular interconnection to provide redundant paths. All major buildings are connected to these three devices via gigabit links (star topology). Approximately 250 edge-switches are located around campus and connected to the gigabit links. The edge-switches provide 100 Mbits/sec links to each individual port on campus. Figure 7.2 provides a summary off the total number of active network ports on campus. Approximately 6,000 network ports serve a population of 2,000 users.

In 2003, IAIT added a third StorageTek L80 robotic tape library to the production computing environment. This tape library utilizes new tape drive technology to store more data (160 GB versus 40 GB) at high transfer rates. At the same time, one of the existing StorageTek L80 tape libraries was upgraded to include a fiber channel interface. The Rose-Hulman computing environment now includes a storage area network with a four TB disk array, two 6-drive, 80-cartridge tape libraries, and six servers. The fiber channel network enables direct data transfers between devices so that the disk array and tape libraries do not have to be directly attached to any of the serves.

IAIT also created a complete mirror system known as the Myers Disaster Recovery Facility. The production database is automatically mirrored to the database server located in Myers Hall. The pre-production database is also located on the Myers database server. The Disaster Recovery Facility includes replicas of the production Kerberos, LDAP and Microsoft Active Directory data stores, thus, enabling a rapid transition to the backup systems. Also in 2003, the Institute upgraded the Olin 203 Public Laboratory with 33 HP xw4000 personal computers. These are well-configured systems that will provide an acceptable computing environment for at least three years.

In 2004, the Institute negotiated a new contract for our Internet connection increasing the bandwidth to 45 Mbs, three times the previous bandwidth. In 2005, the Institute purchased and installed a storage Tek LSA200 disk array. This device added two terrabytes disk capacity, for a total of four terrabytes, for the Microsoft Exchange cluster and other Microsoft Window servers.

Currently all academic buildings are covered by IEEE 802.11b wireless network (over 40 access points). This wireless deployment also incorporates VoIP (voice over internet protocol). In addition, Table B-17 shows the placement of 6036 active fiber connections at the Institution.

Table B-17. Active network connections on campus by building.

Campus Connections	
	Active Network Connections
Crapo Hall	521
Facilities	29
Hadley Hall	39
Hatfield Hall	143
Hulman Union	160
Logan Library	249
Moench Hall	1,480
Myers Hall	215
Observatory	24
Observatory Lodge	18
New Olin Hall	520
Old Olin Hall	520
Public Safety	14
Chi Omega House	10
ROTZ Lab	29
SRC	45
White Chapel	6
Residence Halls - *Totals*	*2,014*
Blumberg Hall	83
Mees Hall	83
Scharpenberg Hall	80
Deming Hall	143
BSB Hall	184
Speed Hall	168
New Residence Hall	662
Skinner Hall	71
New Apartment Res.	540
	6,036

This network is now stable and reliable, allowing faculty to confidently include network-based instructional materials in their courses. In addition to outstanding web pages prepared by individual faculty members, their efforts are now supported by a campus-wide course management system.

d. Project Design Space and Infrastructure

In 1999, the Myers Center for Technological Research with Industry was dedicated. This 40,000 square foot building is used for student project space, with priority given to projects being done

for external clients (e.g. senior engineering design projects). Individual departments are not allowed to "own" space in the building. Rather, space is assigned to project teams on an "as needed" basis. Yearly space allocation is overseen by the associate dean of the faculty. This flexible allocation allows for optimum space utilization as various program enrollments rise and fall.

In addition to flexible project space for both design and construction, the Myers Center houses several support facilities used by students in all departments: a machine shop, rapid prototyping machines (starch and ABS plastic), computer-controlled MTS machine, microscopy lab (x-ray, SEM, fluorescence, optical), and chemical analysis. Here again our guiding philosophy is that of shared resources for maximum utilization.

In addition, it should be noted that four of the under-utilized high bays in the Myers Center have been "capped" providing a second floor in this area. This increased space has been used to provide laboratory/classrooms for the rapidly growing Biomedical Engineering program. When funding is obtained for the planned Biomedical Engineering building, the created space in the Myers Center will become additional project space.

8. Program Criteria

This section will examine how the software engineering program meets the requirements of Criterion 8, as specified in the ABET *Criteria for Accrediting Engineering Programs* approved for the 2006-2007 accreditation cycle.

a. Engineering and Computer Science Topics

The first paragraph of Criterion 8 for software engineering programs reads as follows:

> "The curriculum must provide both breadth and depth across the range of engineering and computer science topics implied by the title and objectives of the program."

The engineering topics involved relate mostly to software engineering and to a lesser degree, computer engineering. Software engineering topics are initially covered as about 40% of the topics in the introductory sequence for SE majors:

- CSSE 120: Fundamentals of Software Development I
- CSSE 220: Fundamentals of Software Development II
- CSSE 230: Fundamentals of Software Development III

The seven 300-level software engineering courses then concentrate on providing greater coverage of topics introduced in the introductory sequence:

- CSSE 371 Software Requirements and Specification
- CSSE 372 Software Project Management
- CSSE 373 Formal Methods in Specification and Design
- CSSE 374 Software Architecture and Design I
- CSSE 375 Software Construction and Evolution
- CSSE 376 Software Quality Assurance
- CSSE 377 Software Architecture and Design II

Finally, the capstone sequence provides practical application of the entire software life cycle to a "real-world" software project:

- CSSE 497 Senior Project I
- CSSE 498 Senior Project II
- CSSE 499 Senior Project III

These 13 courses (4 quarter hours credit each) provide breadth and depth of software engineering topics as specified in the Software Engineering Education Knowledge (SEEK) section of *Software Engineering 2004*, released as part of the IEEE-CS/ACM *Computing Curricula* series. In all, over 430 contact hours of software engineering are used in coverage of all SEEK knowledge areas and knowledge units related to SE, compared to only 275 hours in *Software Engineering 2004.*

Table B-18 (next page) shows the breakdown of those 430 hours compared to the SEEK software engineering knowledge areas and units. As can be seen from this table, Rose-Hulman's software engineering degree program exceeds the SEEK recommendation in every single area, meaning that there is sufficient breadth of software engineering topics in the program. There are also three areas where there is significant depth compared to the SEEK guidelines: formal construction methods (from CSSE 373), software design (CSSE 374 and 377) and evolution (CSSE 375).

Section 8(b) will discuss how these 13 software engineering courses map into the "software systems" requirement defined in the second paragraph on the SE program criteria.

In addition, there are two courses taken by software engineering majors that largely cover computer engineering subject matter:

- Electrical and Computer Engineering (ECE) 130: Introduction to Logic Design
- CSSE 232: Computer Architecture I

The following six courses required of software engineering majors primarily cover computer science topics:

- CSSE 120: Fundamentals of Software Development I
- CSSE 220: Fundamentals of Software Development II
- CSSE 230: Fundamentals of Software Development III
- CSSE 304: Programming Language Concepts
- CSSE 332: Operating Systems
- CSSE 333: Database Systems

These six courses are the same computer science course named requirements in the Computer Science major. As a result, software engineering majors have a solid background in computer science to support their engineering coursework.

b. Software Systems Topics

The second paragraph of Criterion 8 for software engineering programs begins as follows:

> "The program must demonstrate that graduates have: the ability to analyze, design, verify, validate, implement, apply, and maintain software systems…"

Table B-18. Mapping of SEEK software engineering topics to Rose-Hulman software engineering curriculum.

					Contact hours per course								
KA/KU	**Title**	**SEEK Hours**	**RHIT hours**	**RHIT +**	**Intro Seq. (CSSE 120, 220, 230)**	**CSSE 371**	**CSSE 372**	**CSSE 373**	**CSSE 374**	**CSSE 375**	**CSSE 376**	**CSSE 377**	**Senior Project (CSSE 497, 498, 499)**
CMP.ct	Construction Technologies	20	34	+14	20					12			2
CMP.tl	Construction Tools	4	6	+2	4								2
CMP.fm	Formal construction methods	8	20	+12				20					
FND.ec	Engineering economics for software	10	13	+3			5						8
PRF	Professional Practice	35	45	+10	4	8	14	5		4			10
MAA	Software Modeling & Analysis	53	61	+8		20		15	20				6
DES	Software Design	45	80	+35	10	8			15			35	12
VAV	Verification and Validation	42	61	+19	6				5		25	5	20
EVL	Evolution	10	28	+18						20			8
PRO	Process	13	17	+4		1	8			4			4
QUA	Quality	16	30	+14		3					15		12
MGT	Software Management	19	35	+16	6		13						16
	Total	**275**	**430**	**+155**	**50**	**40**	**40**	**40**	**40**	**40**	**40**	**40**	**100**

Notes:
KA = SEEK Knowledge Area
KU = SEEK Knowledge Unit (second level e.g. CMP.ct)
RHIT + = Increase in Rose-Hulman software engineering hours over SEEK minimums

Instruction related to the student's ability to analyze, design, verify, validate, implement, apply, and maintain software systems begins in the Fundamentals of Software Development sequence (CSSE 120, 220 and 230), where students are taught programming, data structures, and algorithms in a manner that will allow them to "scale up" from the smaller software systems that they develop in this sequence to the more complex systems seen in later courses and in the workplace. It is for this reason that an "Introduction to Software Engineering" course such as the one taught in most colleges was found to be superfluous several years ago and subsequently dropped from the list of CSSE courses taught.

Software engineering majors are then exposed to more in-depth coverage of various software systems topics in the 300-level software engineering courses, as shown in Table B-19.

Table B-19. Criterion 8 software systems topics by course.

Analysis	CSSE 371 Software Requirements and Specification CSSE 373 Formal Methods in Specification and Design
Design	CSSE 373 Formal Methods in Specification and Design CSSE 374 Software Architecture and Design I CSSE 377 Software Architecture and Design II
Verification and Validation	CSSE 376 Software Quality Assurance
Implementation, Application and Maintenance	CSSE 375 Software Construction and Evolution

Finally, all of these software systems topics are reinforced and further applied in the capstone sequence (CSSE 497, 498 and 499: Senior Project I, II and III).

c. Supporting Disciplines

The second paragraph of Criterion 8 for software engineering programs continues with:

> "…the ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems…"

Discrete Mathematics is covered in MA 275 and 375 (Discrete and Combinatorial Algebra I and II), and applied in several courses required of SE majors, including CSSE 230 (Fundamentals of Software Development III), which has a significant analysis of algorithms component. CSSE 230 is in turn a prerequisite to all the 300-level software engineering courses, and discrete mathematics is especially important in CSSE 373, the SE formal methods course.

Probability and statistics are covered in MA 381 (Introduction to Probability Theory and Statistics with Applications); this background is especially useful in both CSSE 372 (Software Project Management) and the capstone senior project sequence for cost and effort estimation.

The Fundamentals of Software Development sequence, which covers programming, data structures, and analysis of algorithms, must be completed before taking any of the 300-level

software engineering courses. Other computer science and computer engineering topics (e.g. database systems) are frequently used by teams in the capstone project sequence.

d. Application Domains

Finally, the second and final paragraph of Criterion 8 for software engineering programs concludes as follows:

> "…and the ability to work in one or more significant application domains."

The software engineering curriculum specifies a number of different application domains, of which each graduate must take at least one. The general rules for an application domain are:

- It must be related to a significant software application area.
- It must contain at least 12 hours worth of courses taught outside of the CSSE department.

The process to get a new application domain track approved is

1. Occurs only in the fall term; the Director of Software Engineering will send out an email to faculty and students every year during Week 1.
2. A *Request for New Application Domain* must be filled out by the person who wants the new application domain.
3. The requesting person will need to get approval from the department of the application area *before* submitting the proposal.
4. Proposal must be submitted to Director of Software Engineering by the end of the third week of the fall term.
5. The Director of Software Engineering then consults the SE faculty in Week 4 of the fall. If it is approved, it is submitted to the entire CSSE faculty.
6. The CSSE faculty will consider the proposal for the new main in Week 5 of the fall, in time to students to register in Week 6.
7. The RHIT Curriculum Committee will consider and the proposed new track at their next monthly meeting for final approval

A student may also request a substitution on his or her domain track; this usually because a course that was expected to be taught in the student's senior year is not offered after all. Requests for substitution are accepted every term; they must be made by the Friday before registration week to the Director of Software Engineering, who will accept or reject it by the Monday of registration week.

Below is a list of the 12 current application domain tracks which can be used to satisfy software engineering degree requirements. There have also been tracks which no longer exist due to the elimination of one or more courses in the track; there have also been tracks which have been used by a particular student that includes a "special topics" course i.e. a course that is only being taught on a one-time basis.

Commercial Applications (3 courses, 12 credit hours)

SL 151 Principles of Economics (4)
SL 350 Managerial Accounting (4)
SL 351 Managerial Economics (4) or
VA 453 The Entrepreneur (4) or
SL 354 Microeconomics (4)

Biochemistry (4-5 courses, 16-20 credit hours)

CHEM 202 Engineering Chemistry II (4)
CHEM 230 Intro. to Organic Chem. & Biochemistry (4) or
CHEM 251-252 Organic Chemistry I-II (4 each)
CHEM 330 Biochemistry (4 hours)
CHEM 363 Quantum Chem. & Molecular Spectroscopy (4)

Biomedical (3 courses, 12 credit hours)

BE 310 Analysis of Physiological Systems I (4)
BE 320 Analysis of Physiological Systems II (4)
BE 360 Biomaterials (4)

Electrical Engineering (4 courses, 16 credit hours)

ES 203 Electrical Systems (4)
MA 222 Differential Equations II (4)
ECE 200 Circuits and Systems (4)
ECE 300 Signals and Systems (4)

Ethics and Law of Business (3 courses, 12 credit hours)

RH 101 Introduction to Philosophy (4)
SL 290 Business Law (4)
VA 303 Business and Engineering Ethics (4)

Engineering Management (3 courses, 12 credit hours)

SL 151 Principles of Economics (4)
VA 454 Financial Economics (4)
VA 498 Technology Management and Forecasting (4)

Fundamentals of Engineering (6 courses, 23 credit hours)

MA 222 Differential Equations II (4)
ES 201 Conservation and Accounting (4)
ES 202 Fluid and Thermal Systems (3)
ES 203 Electrical Systems (4)
ES 204 Mechanical Systems (3)
ES 205 Analysis and Design of Engineering Systems (5)

Geography (3 courses, 12 credit hours)

SL 191 Cultural Geography (4)
VA 291 Geography of Europe (4)
GL 391 Geography of Africa & Southwest Asia (4)

International Business and Economics (4 courses, 16 credit hours)

SL 151 Principles of Economics (4)
SL 355 Intermediate Macroeconomics (4)
GL 357 European Economics (4)
GL 359 International Finance (4)

Physical Modeling (4 courses, 16 credit hours)

MA 222 Differential Equations & Matrix Algebra II (4)
MA 371 Linear Algebra (4)
MA 323 Geometric Modeling (4)
MA 433 Numerical Analysis (4)

Scientific Computing (4 courses, 16 credit hours)

MA 222 Differential Equations II (4)
MA 373 Applied Linear Algebra for Engineers (4)
MA 433 Numerical Analysis (4)
MA 439 Mathematics of Image Processing (4)

World Political Studies (3 courses, 12 credit hours, plus one additional requirement)

GL 366 – The European Union (4)
GL 262 – International Relations (4)
SL 363 – European Politics and Government (4)
and some type of international experience related to the domain track (requires CSSE department head approval).

9. General Advanced-Level Program

This section is not applicable, since accreditation of an advanced-level program is not being sought.

Appendix I - Additional Program Information

A. Tabular Data for Program

Table I-1. Basic level Curriculum
Table I-2. Course and Section Size Summary
Table I-3. Faculty Workload Summary
Table I-4. Faculty Analysis
Table I-5. Support Expenditures

Table I-1. Basic-Level Curriculum
Software Engineering

Year; Semester or Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Sciences	Engineering Topics *Check if Contains <u>Significant</u> Design (✓)*	General Education	Other
1st; Fall	CSSE 120 Funds. of Software Development I		4 (√)		
1st; Fall	MA 111 Calculus I	5	( )		
1st; Fall	PH 111 Physics I	4	( )		
1st; Fall	RH 131 Rhetoric and Composition		( )	4	
1st; Fall	CLSK 100 College and Life Skills		( )	1	
1st; Winter	CHEM 201 Engineering Chemistry	4	( )		
1st; Winter	MA 112 Calculus II	5	( )		
1st; Winter	PH 112 Physics II	4	( )		
1st; Winter	HSS (Humanities & Social Science) elective		( )	4	
1st; Spring	ECE 130 Introduction of Logic Design		4 (√)		
1st; Spring	MA 113 Calculus III	5	( )		
1st; Spring	HSS elective		( )	4	
1st; Spring	Science elective (MA, PH or Applied Biology)	4	( )		
2nd; Fall	CSSE 220 Funds. of Software Development II		4 (√)		
2nd; Fall	CSSE 232 Computer Architecture I		4 (√)		
2nd; Fall	MA 221 Diff. Equations and Matrix Algebra I	4	( )		
2nd; Fall	MA 275 Discrete Combinatorial Algebra I	4	( )		
2nd; Winter	CSSE 230 Funds. of Software Development III		4 (√)		
2nd; Winter	CSSE 333 Database Systems		4 ( )		
2nd; Winter	MA 375 Discrete Combinatorial Algebra II	4	( )		
2nd; Winter	Free Elective		( )		4
2nd; Spring	CSSE 304 Programming Language Concepts		4 ( )		
2nd; Spring	MA (Mathematics) elective	4	( )		
2nd; Spring	HSS elective		( )	4	
2nd; Spring	Free Elective		( )		4
3rd; Fall	CSSE 371 Soft. Requirements & Specification		4 ( )		
3rd; Fall	CSSE 373 Formal Methods in Spec. & Design		4 (√)		
3rd; Fall	MA 381 Intro. to Prob. with Statistical Appl.	4	( )		
3rd; Fall	RH 330 Technical Communications		( )	4	
3rd; Winter	CSSE 332 Operating Systems		4 ( )		
3rd; Winter	CSSE 372 Software Project Management		4 ( )		

(continued on next page)

Table 1. Basic-Level Curriculum (continued)

Software Engineering

Year; Semester or Quarter	Course (Department, Number, Title)	Category (Credit Hours)			
		Math & Basic Science	Engineering Topics *Check if Contains <u>Significant</u> Design (✓)*	General Education	Other
3rd; Winter	CSSE 374 Software Architecture & Design I		4 (√)		
3rd; Winter	Free elective		( )		4
3rd; Spring	CSSE 376 Software Quality Assurance		4 ( )		
3rd; Spring	CSSE 377 Software Architecture & Design II		4 (√)		
3rd; Spring	HSS elective		( )	4	
3rd; Spring	Free elective		( )		4
4th; Fall	CSSE 375 Construction and Evolution		4 ( )		
4th; Fall	CSSE 497 Senior Project I		4 (√)		
4th; Fall	HSS elective		( )	4	
4th; Fall	Free elective		( )		4
4th; Winter	CSSE 498 Senior Project II		4 (√)		
4th; Winter	(restricted) CSSE elective		4 ( )		
4th; Winter	HSS elective		( )	4	
4th; Winter	Free elective		( )		4
4th; Spring	CSSE 499 Senior Project III		4 (√)		
4th; Spring	HSS elective		( )	4	
4th; Spring	Free elective		( )		4
			( )		
			( )		
			( )		
TOTALS-ABET BASIC-LEVEL REQUIREMENTS		51	76	37	28
OVERALL TOTAL FOR DEGREE	192 quarter hours				
PERCENT OF TOTAL					
Totals must satisfy one set	Minimum quarter credit hours	48 hrs	72 hrs		
	Minimum percentage	25%	37.5 %		

Note about Application Domain Track Courses: For the places in the degree plan where it says “Domain track elective” or “Domain track or free elective”, this table merely has “free elective” in order emphasize that the domain track courses can also elsewhere in the degree plan, depending on the type of course. For instance, a math course in a particular application domain track can be used as the MA elective.

Table I-2. Course and Section Size Summary

Software Engineering

Course No.	Title	No. of Sections offered in Current Year	Avg. Section Enrollment	Type of Class[1]			
				Lecture	Laboratory	Recitation	Other
CSSE 120	Funds. of Soft. Development I	7	18	50%	50%		
CSSE 220	Funds. of Soft. Development II	4	14	50%	50%		
CSSE 221	Funds. of Soft. Dev. Honors	2	21	50%	50%		
CSSE 230	Funds. of Soft. Development III	1	26	50%	50%		
CSSE 232	Computer Architecture I	4	25	50%	50%		
CSSE 304	Programming Lang. Concepts	3	18	100%			
CSSE 332	Operating Systems	5	21	50%	50%		
CSSE 333	Database Systems	2	22	50%	50%		
CSSE 351	Computer Graphics	2	18	100%			
CSSE 371	Software Reqs. & Specification	2	27	100%			
CSSE 372	Software Project Management	2	29	100%			
CSSE 373	Formal Methods for Spec & Des.	1	25	100%			
CSSE 374	Soft. Architecture & Design I	1	22	100%			
CSSE 375	Soft. Construction & Evolution	1	19	100%			
CSSE 376	Software Quality Assurance	1	24	100%			
CSSE 377	Soft. Architecture & Design II	1	22	100%			
CSSE 404	Compiler Construction	1	12	100%			
CSSE 413	Artificial Intelligence	2	15	100%			
CSSE 432	Computer Networks	1	29	100%			
CSSE 433	Advanced Database Systems	0	--	100%			
CSSE 442	Computer Security	1	21	100%			
CSSE 451	Advanced Computer Graphics	1	21	100%			

Enter the appropriate percent for each type of class for each course (e.g., 75% lecture, 25% recitation).

Table I-2. Course and Section Size Summary (continued)

Software Engineering

Course No.	Title	No. of Sections offered in Current Year	Avg. Section Enrollment	Type of Class[1]			
				Lecture	Laboratory	Recitation	Other
CSSE 453	Topics in Artificial Intelligence	0	--	100%			
CSSE 461	Computer Vision	0	--	100%			
CSSE 481	Web-Based Information Systems	1	28	100%			
CSSE 497	Senior Project I	4	11		100%		
CSSE 498	Senior Project II	3	15		100%		
CSSE 499	Senior Project III	4	11		100%		

Enter the appropriate percent for each type of class for each course (e.g., 75% lecture, 25% recitation).

Table I-3. Faculty Workload Summary
Software Engineering

Faculty Member (Name)	FT or PT (%)	Classes Taught (Course No./Credit Hrs.) Term and Year[1]	Total Activity Distribution[2]		
			Teaching	Research	Other[3]
Claude Anderson	100	None			100% (Sabbactical)
Mark Ardis	100	CSSE 373 (4), 497 (4), 497 (4)/Fall 2005-06 CSSE 374 (4), 498 (4), 498 (4)/Winter 2005-06 CSSE 376 (4), 497 (4), 499 (4), 499 (4), Spring 2005-06	80%	20%	
Donald Bagert	100	CSSE 371 (4), CSSE 371 (4)/Fall 2005-06 CSSE 372 (4), CSSE 372 (4)/Winter 2005-06 CSSE 220 (4), CSSE 375 (4)/Spring 2005-06	46.7%	20%	33.3% (Administration)
Matthew Boutell	100	CSSE 120 (4)/Fall 2005-06 CSSE 230 (4), 230 (4)/Winter 2005-06 CSSE 230 (4), 490 (4)/Spring 2005-06	80%	20%	
Stephen Chenoweth	100	CSSE 413(4),413 (4),497(4),499(4)/Fall 2005-06 CSSE 333 (4), 498 (4)/Winter 2005-06 CSSE 377 (4), 499 (4)/Spring 2005-06	80%	20%	
Archana Chidanandan	100	CSSE 232 (4), 232 (4)/Fall 2005-06 CSSE 332 (4), 332 (4)/Winter 2005-06 CSSE 332 (4), 432 (4)/Spring 2005-06	80%	20%	
Curtis Clifton	100	CSSE 120 (4) /Fall 2005-06 CSSE 333 (4), 333 (4)/Winter 2005-06 CSSE 120 (4), 404 (4)/Spring 2005-06	80%	20%	
Lisa Kaczmarczyk	100	CSSE 221 (4)/Fall 2005-06 CSSE 120 (4), 120 (4)/Winter 2005-06 CSSE 120 (4), 490 (4)/Spring 2005-06	80%	20%	

1. Indicate Term and Year for which data apply.
2. Activity distribution should be in percent of effort. Faculty member's activities should total 100%.
3. Indicate sabbatical leave, etc., under "Other."

Table I-3. Faculty Workload Summary (continued)
Software Engineering

Faculty Member (Name)	FT or PT	Classes Taught (Course No./Credit Hrs.) Term and Year[1]	Total Activity Distribution[2]		
Cary Laxer	100	CSSE 351(4), 494 (4)/Fall 2005-06 CSSE 451(4), 490(4), 495(4), 496(4)/Winter 2005-06 CSSE 325 (4), 351 (4)/Spring 2005-06	13.3%	20%	66.7% (Administration)
J.P. Mellor	100	CSSE 491 (4)/Winter 2005-06	5%		95% (Sabbactical)
Larry Merkle	100	CSSE 232 (4), 494 (4)/Fall 2005-06 CSSE 232 (4), 494 (4)/Winter 2005-06 CSSE 332 (4), 442 (4)/Spring 2005-06	46.7%	20%	33.3% (Administration)
David Mutchler	100	CSSE 120 (4), 221 (4), 494 (4)/Fall 2005-06 CSSE 220 (4), 220 (4), 495 (4)/Winter 2005-06 CSSE 332 (4), 491 (4), 496 (4)/Spring 2005-06	80%	20%	
Michael Wollowski	100	CSSE 220 (4), 481 (4), 494 (4)/Fall 2005-06 CSSE 474(4),474(4),490(4),491(4),495(4)/Winter 2005-06 CSSE 304 (4), 304 (4), 304 (4), 491 (4), 496 (4)/Spring 2005-06	80%	20%	

1. Indicate Term and Year for which data apply.
2. Activity distribution should be in percent of effort. Faculty member's activities should total 100%.
3. Indicate sabbatical leave, etc., under "Other."

Table I-4. Faculty Analysis
Software Engineering

Name	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			State in which Registered	Level of Activity (high, med, low, none)		
					Govt./ Industry Practice	Total Faculty	This Institution		Professional Society (Indicate Society)	Research	Consulting /Summer Work in Industry
Claude Anderson	Full	FT	Ph.D.	Illinois-Urbana, 1981	3	25	18		ACM – low	low	high
Mark Ardis	Full	FT	Ph.D.	Univ. of Maryland, 1980	14	19	6		IEEE – med ACM – low ASEE – low	high	none
Donald Bagert	Full	FT	Ph.D.	Texas A&M, 1986	0	27	4	Texas	IEEE – high ASEE – med ACM – low NSPE – low	high	none
Matthew Boutell	Asst.	FT	Ph.D.	Univ. of Rochester, 2005	0	2.5 (+ 6 yrs high School)	1		IEEE – med	high	high
Stephen Chenoweth	Assoc.	FT	Ph.D.	Wright State Univ., 1990	28	3	3		IEEE – low ACM – low ASEE – low	med	low

Instructions: Complete table for each member of the faculty of the program. Use additional sheets if necessary. Updated information is to be provided at the time of the visit. The level of activity should reflect an average over the current year (year prior to visit) plus the two previous years.

Table I-4. Faculty Analysis (continued)
Software Engineering

Name	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			State in which Registered	Level of Activity (high, med, low, none)		
					Govt./ Industry Practice	Total Faculty	This Institution		Professional Society (Indicate Society)	Research	Consulting /Summer Work in Industry
Archana Chidanandan	Asst.	FT	Ph.D.	Univ. of La. At Lafayette, 2004	1	3	3		IEEE – med ACM – med	high	none
Curtis Clifton	Asst.	FT	Ph.D.	Iowa State Univ., 2005	6	1	1		ACM – med IEEE – low ASEE – low	high	none
Lisa Kaczmarczyk	Asst.	FT	Ph.D.	Univ. of Texas At Austin, 2005	7	13	1		ACM – high Cognitive Science Soc. – med IEEE – low	high	none
Cary Laxer	Full	FT	Ph.D.	Duke, 1980	0	25	25		ACM – high IEEE – low ASEE – low	med	none
J.P. Mellor	Assoc.	FT	Ph.D.	MIT, 2000	11	7	7		IEEE – low ACM – low	high	High

Instructions: Complete table for each member of the faculty of the program. Use additional sheets if necessary. Updated information is to be provided at the time of the visit. The level of activity should reflect an average over the current year (year prior to visit) plus the two previous years.

Table I-4. Faculty Analysis (continued)

Software Engineering

Name	Rank	FT or PT	Highest Degree	Institution from which Highest Degree Earned & Year	Years of Experience			State in which Registered	Level of Activity (high, med, low, none)		
					Govt./ Industry Practice	Total Faculty	This Institution		Professional Society (Indicate Society)	Research	Consulting /Summer Work in Industry
Larry Merkle	Asst.	FT	Ph.D.	Air Force Inst. Of Tech., 1996	6	9	4		ACM – high IEEE – high ASEE – low AAAI – low	high	medium
David Mutchler	Full	FT	Ph.D.	Duke Univ., 1986	1	20	12		ACM – low AAAI – low	med	none
Michael Wollowski	Assoc.	FT	Ph.D.	Indiana Univ., 1998	0	11	7		ACM – med	med	none

Instructions: Complete table for each member of the faculty of the program. Use additional sheets if necessary. Updated information is to be provided at the time of the visit. The level of activity should reflect an average over the current year (year prior to visit) plus the two previous years.

Table I-5. Support Expenditures

Software Engineering

Fiscal Year	2004 (prior to previous year)	2005 (previous year)	2006 (current year)	2007 (year of visit)
Expenditure Category				
Operations[1] (not including staff)	29,368	29,987	30,900	32,272
Travel[2]	12,874	16,685	18,750	18,750
Equipment[3]				
Institutional Funds	-	-	-	-
Grants and Gifts[4]	-	-	7,800	-
Graduate Teaching Assistants	-	-	-	-
Part-time Assistance[5] (other than teaching)	6,273	2,993	7,500	5,000

Instructions:

Report data for the engineering program being evaluated. Updated tables are to be provided at the time of the visit.

Column 1: Provide the statistics from the audited account for the fiscal year completed 2 years prior to the current fiscal year.
Column 2: Provide the statistics from the audited account for the fiscal year completed prior to your current fiscal year.
Column 3: This is your **current fiscal year** (when you will be preparing these statistics). Provide your preliminary estimate of annual expenditures, since your current fiscal year presumably is not over at this point.
Column 4: Provide the budgeted amounts for your next fiscal year to cover the fall term when the ABET team will arrive on campus.

Notes:

1. General operating expenses to be included here.
2. Institutionally sponsored, excluding special program grants.
3. Major equipment, excluding equipment primarily used for research. Note that the expenditures under "Equipment" should total the expenditures for Equipment. If they don't, please explain.
4. Including special (not part of institution's annual appropriation) non-recurring equipment purchase programs.
5. Do not include graduate teaching and research assistant or permanent part-time personnel.

Appendix I

(continued)

B. Course Syllabi

Computer Science and Software Engineering	CSSE 120 Fundamentals of Software Development 1

Catalog Description

This course introduces students to current practices of producing object-oriented software. Using a disciplined design process, students explore multi-threaded event-driven programming, the development of graphical user interfaces, and interaction among objects. Students design and implement software individually, in small groups, and in a challenging multi-week team project. This course presumes no prior programming experience.

Instructor(s)

Boutell, Clifton, Kaczmarczyk, Mutchler

Required/Elective

Required of majors in Computer Science, Software Engineering, Computer Engineering, Electrical Engineering, Physics, Mathematics and Economics

Class/Laboratory Schedule

Three 105-minute sessions per week

Prerequisites

None

Textbook(s)

- Required textbook: *Interactive Programming in Java*, by Lynn Andrea Stein, 2004. Online and free to students.
- Supplementary textbook: *Big Java*, by Cay Horstmann, J. Wiley, 2006.

Course Objectives

After successful completion of this course, students will:

1. Analyze and explain the behavior of simple programs involving:
    a. Fundamental non-object-oriented programming constructs, including:
        - Basic syntax and semantics of a higher-level language
        - Variables, types, expressions, and assignment
        - Simple I/O
        - Conditional and iterative control structures
        - Functions and parameter passing
        - Arrays
    b. Fundamental object-oriented programming constructs, including:
        - Basic syntax and semantics of a class
        - Fields, constructors and methods
        - The distinction between primitive types/variables and non-primitive types/variables
        - The relationship between the static structure of the class and the dynamic structure of the instances of the class (e.g. via Java's *new* operator)
        - Visibility modifiers (e.g. Java's *public*, *protected* and *private*) and mutability

modifiers (e.g. Java's *final*)
    - The distinction between a static method/field and a non-static method/field
    - Implementing an interface
    - Extending a class
    - Responding to an exception
2. Design, implement, debug and informally test simple programs in an object-oriented programming language, including programs that:
    a. Use the above fundamental programming constructs
    b. Support a graphical user interface developed by using a GUI toolkit
    c. Respond to user events
    d. Use threads
    e. Include "is-a" relationships among objects using a class hierarchy and inheritance
    f. Use large-scale API packages
3. Apply decomposition techniques, including:
    a. Using structured (functional) decomposition (breaking a function/program into subfunctions)
    b. Encapsulating behaviors and data into classes
    c. Using inheritance
4. Explain techniques that scale up to larger software development projects (and explain why they scale up), including:
    a. Pair programming
    b. Structured documentation (e.g., javadoc comments)
    c. Implementing by using stubs
    d. Implementing by using an iterative enhancement plan
    e. Using code conventions (e.g., Sun's code conventions)
    f. Decomposition techniques listed above
    g. Using an integrated development environment
    h. Using UML class diagrams
    i. Unit testing using a tool like JUnit
    j. Using simple metrics to evaluate quality of code
    k. Using simple team building and management techniques like brainstorming, roles and task lists
5. Work effectively as members of a small team

Course Topics

Software development practices, fundamental object-oriented and non-object-oriented programming constructs.

Lab Topics

Mixed in with lecture.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d, e and k.

Program Outcomes

This course addresses SE Program Outcome SE1.

Prepared by

David Mutchler, CSSE 120 coordinator; additional data inserted by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 220 Fundamentals of Software Development II

Catalog Description

This course reinforces and extends students' understanding of current practices of producing object-oriented software. Students extend their use of a disciplined design process to include testing and space/time efficiency analysis. Students gain a deeper understanding of concepts from CSSE 120, including the use of inheritance, interfaces, polymorphism, abstract data types and encapsulation to enable software reuse and assist in software maintenance. This course introduces networking and database applications, and recursion as an example of functional programming. Students design and implement software individually, in small groups, and in a challenging multi-week team project.

Instructor(s)

Wollowski, Mutchler, Bagert

Required/Elective

Required.

Class/Laboratory Schedule

Three 105 minute sessions each week, split into sections of lectures, demonstrations, student exercises, group and team work.

Prerequisites

CSSE 120 – Fundamentals of Software Development I

Textbook(s)

Weiss, Mark Allen: *Data Structures & Problem Solving using Java*, Third Edition
Gittleman, Art: *Advanced Java: Internet Applications*, Second Edition

Course Objectives

After successful completion of this course, students will be able to:

1. Develop software that incorporates the following techniques:
    - Inheritance
    - Interfaces
    - Polymorphism
    - Casting
    - Function objects
    - Generics
    - Collections
2. Develop software in each of the following application areas:
    - Software that interacts with a database
    - Software that communicates over a network
    - Software that uses a graphical user interface

3. Perform the following steps of the software development cycle effectively:
    - Requirements elicitation
    - Design expressed as UML class diagrams
    - Implementation using an iterative enhancement plan and stubs/documentation before coding
    - Unit, system, and acceptance testing
4. Analyze the asymptotic worst, best, and average case run times of simple algorithms.
5. Select basic data structures (e.g, arrays, sequential lists, linked lists, stacks, queues, hash tables and trees) based on the time and space complexity of typical operations.
6. Develop software that uses recursive methods and data structures.

Course Topics

- Inheritance and Polymorphism
- Static overloading
- Dynamic binding
- Interfaces and abstract classes
- Method abstraction
- Functors
- Generic algorithms
- Iterators
- Recursion
- Analysis of algorithms
- Searching and Sorting
- Data Structures (arrays, stacks, queues, linked lists, binary trees)
- Software engineering: Requirements elicitation, use cases, acceptance testing
- Database connectivity
- Networking

Lab Topics – see above

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d, e and k.

Program Outcomes

This course addresses SE Program Outcomes SE1 and SE4

Professional Component

This course provides 4 credits under engineering topics and contains significant design.

Prepared by

Michael Wollowski, 04/01/06; additional data inserted by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 221 Fundamentals of Software Development Honors

Catalog Description

This course is intended for students who have sufficient programming experience to warrant placement in an accelerated course covering the topics from CSSE 120 and CSSE 220. This course will satisfy the prerequisite requirements for courses that have CSSE 220 as a prerequisite. Students who receive a penalty grade in CSSE 221 may grade replace it with CSSE 220.

Students with a score of 4 or 5 on the Computer Science A or AB Advanced Placement exam may enroll in CSSE 221. Upon successful completion of CSSE 221 students will also be awarded 4 credits for CSSE 120. Furthermore, students with a score of 4 or 5 on the Computer Science AB Advanced Placement exam who complete CSSE 221 with a grade of C or better will be awarded a further 4 credits for CSSE 230.

Instructor(s)

Mutchler, Kaczmarczyk

Required/Elective

Is an accelerated course which can be used in place of required courses

Class/Laboratory Schedule

Three 105-minute sessions per week

Prerequisites

This course is intended for students who have sufficient programming experience to warrant placement in an accelerated course covering the topics from CSSE 120 and CSSE 220. Students with a score of 4 or 5 on the Computer Science A or AB Advanced Placement exam are normally enrolled in CSSE 221. Other students may enroll with the instructor's permission.

Textbook(s)

- Required textbook: *Interactive Programming in Java*, by Lynn Andrea Stein, 2004. Online and free to students.
- Required textbook: *Big Java*, by Cay Horstmann, J. Wiley, 2006.

Course Objectives

After successful completion of this course, students will:

1. Meet all the learning outcomes of CSSE 120.
2. Meet all the learning outcomes of CSSE 220

Course Topics

- **From CSSE 120:** Software development practices, fundamental object-oriented and non-object-oriented programming constructs. Developing software by working in teams.
- **From CSSE 220:** Inheritance; Polymorphism; Static overloading; Dynamic binding; Interfaces and abstract classes; Method abstraction; Functors; Generic algorithms; Iterators; Recursion; Analysis of algorithms; Searching; Sorting; Data structures, including: arrays, ArrayLists, Stacks, Queues, Linked Lists, Vectors, binary trees; The following concepts of software engineering: Requirements elicitation, use cases, acceptance testing; Databases connectivity; Networking.

Lab Topics

None.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d, e and k.

Program Outcomes

This course addresses SE Program Outcomes SE1 and SE4.

Professional Component

This course provides 4 credits under engineering topics and contains significant design.

Prepared by

David Mutchler, CSSE 221 coordinator; additional data added by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 230 Fundamentals of Software Development III

Catalog Description

This course reinforces and extends students' understanding of current practices of producing object-oriented software. Students extend their use of a disciplined design process to include formal analysis of space/time efficiency analysis (including recurrence relations and the big-Oh, Omega and Theta notations) and formal proofs of correctness (including mathematical induction). Students gain a deeper understanding of ideas from CSSE 220, including implementations of abstract data types by linear and non-linear data structures. This course introduces the use of randomized algorithms. Students design and implement software individually, in pairs, and in a team project.

Instructor(s)

Boutell

Required/Elective

Required.

Class/Laboratory Schedule

Three 105 minute sessions each week, split into sections of lectures, demonstrations, student exercises, group and team work.

Prerequisites

CSSE 220 or 221: Fundamentals of Computing II
MA 275: Discrete Mathematics I

Textbook(s)

Weiss, Mark Allen: *Data Structures & Problem Solving using Java*, Third Edition

Course Objectives

After successful completion of this course, students will be able to:

1. Describe the classical data structures (list, stack, queue, tree, priority queue, hash table, graph, set, dictionary) and understand the issues involved in implementation choices for each.
2. Describe the brute force, divide-and-conquer, greedy, dynamic programming, branch-and-bound and backtracking strategies, and recognize problems for which each strategy provides an appropriate solution.
3. Analyze and prove the asymptotic worst, best, and average case run times of algorithms, including non-trivial recursive algorithms, using Big O, little o, Omega, and Theta notation.
4. Analyze and prove lower-bound results on problems like comparison-based sorting and understand the implications of lower bound results.
5. Understand and analyze classical sorting, graph and tree-balancing

algorithms.

6. Prove the correctness of simple algorithms, especially those involving loops and recursion.
7. Work effectively as part of a team.
8. Independently analyze problems, and design and implement solutions.

Course Topics

- Binary Search Trees
- Tree iterators
- AVL trees
- Red Black trees
- AA trees
- Splay trees
- Hash tables
- Heaps
- Graph traversal algorithms
- Huffman codes
- Advanced sorting algorithms
- Sorting lower bounds
- Analysis of algorithms

Lab Topics

<see above>

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d, e and k.

Program Outcomes

This course addresses SE Program Outcome SE1 and Institutional Outcome RH7.

Professional Component

This course provides 4 credits under engineering topics and contains significant design.

Prepared by

Michael Wollowski, 04/01/06; additional data added by Don Bagert 06/09/06

Computer Science and Software Engineering	CSSE 232 Computer Architecture I

Catalog Description

Computer instruction set architecture and implementation. Specific topics include historical perspectives, performance evaluation, computer organization, instruction formats, addressing modes, computer arithmetic, ALU design, floating-point representation, single-cycle and multi-cycle data paths, and processor control. Assembly language programming is used as a means of exploring instruction set architectures. The final project involves the complete design and implementation of a miniscule instruction set processor.

Instructor(s)

Larry Merkle, Archana Chidanandan

Required/Elective

Required.

Class/Laboratory Schedule

Three 100-minute lecture and laboratory sessions each. The laboratory component of the class is incorporated into the regular lecture session. Students bring their laptop to class.

Prerequisites

CSSE 120, ECE 130

Textbook(s)

Patterson and Hennessy, "Computer Organization and Design," 3rd ed., Morgan Kaufman

Course Objectives

After successful completion of this course, students will:

- Apply the principle of abstraction in analysis and design problems.
- Explain the binary representation of various forms of data.
- Given the syntax and semantics for an assembly language, design, implement, test, and debug simple programs in that language that involve arithmetic operations, input and output, various control structures (including selection, iteration, and recursive procedures), and interrupts.
- Design an instruction set architecture that is appropriate for a given application, taking into consideration key computer organization design principles.
- Design a hardware implementation of an instruction set architecture, and use modern computer aided design tools to model, simulate, test, and debug that implementation.
- Analyze the performance of an instruction set architecture and

implementation in terms of metrics such as CPU execution time, instruction count, cycles per instruction (CPI), clock frequency, throughput, and response time.
- Work effectively as members of a team.
- Communicate effectively in both written and verbal form.

Course Topics

- MIPS assembly language and machine language
- Data representation
- Computer arithmetic
- Register Transfer Language
- Datapath design
- Control unit design
- ALU design
- Evaluation of the performance of a computer architecture
- Use of SPIM and Xilinx software tools

Lab Topics

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d, e and g.

Program Outcomes

This course addresses SE Program Outcomes SE1, SE4 and SE5.

Professional Component

This course provides 4 credits under engineering topics and contains significant design.

Prepared by

Archana Chidanandan, 7 Oct 2005; additional data added by Don Bagert 9 June 2006

Computer Science and Software Engineering	CSSE 304 Programming Language Concepts

Catalog Description

Syntax and semantics of programming languages. Grammars, parsing, data types, control flow, parameter passing, run-time storage management, binding times, functional programming and procedural abstraction, syntactic extensions, continuations, language design and evaluation. Students will explore several language features by writing an interpreter that implements them. individually, in pairs, and in a team project.

Instructor(s)

Wollowski

Required/Elective

Required.

Class/Laboratory Schedule

Four 50 minute sessions each week.

Prerequisites

CSSE 230 (Note that this course has MA215 as a prerequisite).
MA 315 recommended.

Textbook(s)

Essentials of Programming Languages Second Edition by Friedman, Wand, and Haynes.
The Scheme Programming Language, Third Edition by R. Kent Dybvig

Course Objectives

After successful completion of this course, students will be able to:

1. Design, implement, debug, and test complex programs in a functional programming language
2. Understand the role of:
    - Syntax specification
    - Syntax and semantics
    - Environments and continuations
    - Types
    - Abstraction
3. Understand the process of interpreting a program
4. Understand the relationship between syntax specification and implementation
5. Design, implement, debug, and test a complex interpreter

Course Topics

- Backus-Naur Form
- Creating and evaluating functions in Scheme
- Lexical binding and scope
- Free and bound variables
- Abstract data types
- Parsing
- Environments
- Interpreters
- Lambda expressions
- Continuations
- Continuation passing style

Lab Topics

None

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Michael Wollowski, 04/01/06; additional data added by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 332 Operating Systems

Catalog Description

Students learn fundamental concepts of modern operating systems by studying how and why operating systems have evolved. Topics include CPU scheduling, process synchronization, memory management, file systems, I/O systems, privacy and security, and performance evaluation. Students implement parts of an operating system as a means of exploring the details of some of these topics.

Instructor(s)

Archana Chidanandan, Larry Merkle

Required/Elective

Required.

Class/Laboratory Schedule

Three 100-minute lecture and laboratory sessions. The laboratory exercises are incorporated into the lecture sessions. Students bring their laptops to class. The class is held in D219 which is equipped with dual-bootable computers.

Prerequisites

CSSE 220, CSSE 232

Textbook(s)

W. Stallings, "Operating Systems: Internals and Design Principles", 5th ed., Prentice Hall

Course Objectives

After successful completion of this course, students will:

- Explain the objectives and functions of modern operating systems.
- Describe how operating systems have evolved over time from primitive batch systems to sophisticated multiuser systems.
- Analyze the tradeoffs inherent in operating system design.
- Explain the hardware support necessary to implement protection mechanisms in operating systems.
- Analyze the performance of preemptive and non-preemptive scheduling algorithms.
- Explain how processes are created, managed and deleted by operating systems.
- Explain the need for process synchronization and demonstrate the proper use of common synchronization techniques.
- Explain memory hierarchy, including virtual memory, caching, paging and segmentation, and analyze memory performance and cost-performance tradeoffs.

- Explain the different approaches to file organization and their strengths and weaknesses.
- Design, implement, test, and debug software using a high-level language that uses explicit pointers, as well as explicit static and dynamic memory allocation.
- Work effectively in teams.
- Communicate effectively in writing.

Course Topics

- Introduction to UNIX and C programming
- History of operating systems
- Process model and thread models
- Concurrency issues such as synchronization, mutual exclusion, and deadlock.
- File systems
- Memory management
- Virtual memory
- Process scheduling
- Disk and I/O scheduling

Lab Topics

None.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d and e.

Program Outcomes

This course addresses SE Program Outcomes SE1, SE4 and SE5.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Archana Chidanandan, 7th October 2005

Computer Science and Software Engineering	CSSE 333 Database Systems

Catalog Description

Relational database systems, with emphasis on entity relationship diagrams for data modeling. Properties and roles of transactions. SQL for data definition and data manipulation. Use of contemporary API's for access to the database. Enterprise examples provided from several application domains. The influence of design on the use of indexes, views, sequences, joins, and triggers. Physical level data structures: B+ trees and RAID. Survey of object databases.

Instructor(s)

Clifton, Chenoweth

Required/Elective

Required.

Class/Laboratory Schedule

- Approx three 50 minute lecture / discussions per week
- Approx three 55-minute in-class lab and project sessions per week (the class is taught in a 3 by 1:45 studio format each week)
- Team presentation of projects (5 x 50 minutes each)
- Lab and project work also done outside class, average from 2004-5 class was approx 40 hours per student total.

Prerequisites

CSSE 220 – Fundamentals of Software Development II, with a co-requisite of CSSE 230 –Fundamentals of Software Development III.

Textbook(s)

Fundamentals of Database Systems, 4/E, by Ramez Elmasri and Shamkant B. Navathe. Addison-Wesley, 2004, ISBN: 0-321-12226-7. **Required.**

Self-Paced Training Kit: Microsoft SQL Server 2000 System Administration, Microsoft Press, ISBN: 0735619611, and Self-Paced Training Kit: Microsoft SQL Server 2000 Database Design and Implementation, Microsoft Press (this one's now out of print, replacement text is ISBN: 073561248X). **Both Suggested.**

Course Objectives

After successful completion of this course, students will:

- An understanding of information models and systems
- A thorough understanding of Entity-Relationship model and Relational Model
- An awareness of other data modeling paradigms (such as legacy systems, object-oriented databases, etc.)
- An ability to design relational databases from problem statements
- An understanding of the basic aspects of physical database design

- An understanding and usage of a SQL-based query languages to access and modify data
- An ability to use database programming techniques such as stored procedures, triggers, etc.
- An understanding of information privacy, integrity, security, and preservation
- A basic understanding of Online Analytical Processing (OLAP) systems and Data Warehousing

Course Topics

Database Theory

- Modeling (ER, Relational, Normalization, Dependencies)
- Relational Algebra
- SQL, Transact-SQL, Embedded SQL
- Constraints
- Store Procedures
- Authorization
- Transactions
- Object Orientation
- XML
- OLAP/Data warehousin

Database Practice

- Managing DB Files, Security
- Admin Tasks, Backing Up, Restoring
- Monitoring Performance
- Implementing DB Design
- Creating and Managing Databases, Data Types, Tables
- Data Integrity
- Indexes
- Views
- Stored Procedures
- Triggers
- Retrieving, Grouping
- Joining Tables
- Modifying Data

Lab Topics

None

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Steve Chenoweth; additional data added by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 351 Computer Graphics

Catalog Description

Computer graphics algorithms, hardware and software. Line generators, affine transformations, line and polygon clipping, interactive techniques, perspective projection, solid modeling, hidden surface algorithms, lighting models, shading, and graphics standards. Programming assignments and a final project are required.

Instructor(s)

Laxer, Mellor

Required/Elective

Elective

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

CSSE 220, Fundamentals of Software Development II
MA 221, Differential Equations and Matrix Algebra I

Textbook(s)

Computer Graphics with OpenGL (Third Edition) by Hearn and Baker. Pearson Prentice Hall, 2004.

Course Objectives

After successful completion of this course, students will be able to:

- develop interactive computer graphics applications using a modern graphics API
- model geometric objects, curves, and surfaces
- project a three-dimensional scene onto a two-dimensional viewing surface
- apply lighting and shading to a scene
- describe the mathematical foundations of computer graphics (e.g. Bresenham's algorithm, transformation matrices, projection matrices)
- apply a texture map to an object

Course Topics

Graphics systems
OpenGL
Graphics primitives and attributes
Geometric transformations
Interaction
Two- and three-dimensional viewing

Shading, lighting, and texture mapping
Curve and surface design

Lab Topics

None

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Cary Laxer, 3/28/06; additional data added by Don Bagert 06/09/06

Computer Science and Software Engineering	CSSE 371 Software Requirements and Specification

Catalog Description

Basic concepts and principles of software requirements engineering, its tools and techniques, and methods for modeling software systems. Topics include requirements elicitation, prototyping, functional and non-functional requirements, object-oriented techniques, and requirements tracking.

Instructor(s)

Don Bagert

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures each week

Prerequisites

CSSE 230, Fundamentals of Software Development III

Textbook(s)

- *Managing Software Requirements: A Use Case Approach*, Second Edition, by Dean Leffingwell and Don Widrig, ISBN 0-321-12247-X
- *Interaction Design: beyond human-computer interaction* by Jennifer Preece, Yvonne Rogers and Helen Sharp, ISBN 0-471-49278-7)

Course Objectives

After successful completion of this course, students will:

Students who complete this course should be able to:

1. Understand the role of requirements engineering and its process
2. Use analysis techniques to develop a problem statement
3. Use multiple techniques to elicit requirements from stakeholders
4. Take the elicited requirements and develop a specification with functional and non-functional requirements
5. To be able to negotiate with the client and other stakeholders regarding priorities and scope
6. Use quality assurance techniques to verify that requirements are: verifiable, traceable, measurable, testable, accurate, unambiguous, consistent, and complete
7. Know how to manage requirements
8. To be able to develop and use user interface prototypes to validate requirements

Course Topics

- Problem Analysis
- Requirements Elicitation
- Development of Software Requirements
- Managing Scope of Requirements
- Quality Assurance of Requirements
- Requirements Management
- User Analysis
- User Interface Protoptyping

Lab Topics

None.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, d and e.

Program Outcomes

This course addresses SE Program Outcomes SE1, SE2, SE4 and SE6, and Institutional Outcome RH2.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Don Bagert; March 28, 2006 (additional data added on June 9, 2006)

Computer Science and Software Engineering	CSSE 372 Software Project Management

Catalog Description

Major issues and techniques of project management. Project evaluation and selection, scope management, team building, stakeholder management, risk assessment, scheduling, quality, rework, negotiation, and conflict management.

Instructor(s)

Don Bagert

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures each week

Prerequisites

CSSE 230, Fundamentals of Software Development III

Textbook(s)

Rapid Development by Steve McConnell, Microsoft Press, 1996, ISBN 1-55615-900-5.

Course Objectives

After successful completion of this course, students will:

1. Explain several different lifecycle models for software engineering
2. Explain the fundamentals of software project management, including organizational structures and roles
3. Prepare a software project plan, including risk analysis
4. Lead or participate on a software project team, including sharing of responsibilities, building consensus, developing plans of action, and providing constructive feedback to peers
5. Plan, facilitate and participate in effective meetings
6. Participate as a member of a team that successfully monitors and controls a software project using traditional metrics
7. Conduct a software project retrospective
8. Explain the role of software engineering ethics and professional practices within the global society
9. Explain the role of motivation, satisfaction and lifelong learning in software professional careers
10. Use standard configuration management methods
11. Apply their understanding of configuration management in a software project team

Course Topics

- Software Methodologies
- Project Planning and Scheduling
- Risk Analysis
- Effective Project Meetings
- Project Team Coordination
- Project Tracking and Monitoring
- Software Metrics Selection and Usage
- Software Configuration Management
- Software Engineering Professionalism
- Software Engineering Ethics
- Software Engineering Professional Development

Lab Topics

None

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d, e, f, g, h, i, j and k.

Program Outcomes

This course addresses SE Program Outcomes SE1, SE2, SE4, SE5, SE7, SE8 and SE9, and Institutional Outcomes RH1 and RH5.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Don Bagert; March 28, 2006; modified June 9, 2006

Computer Science and Software Engineering	CSSE 373 Formal Methods in Specification and Design

Catalog Description

Introduction to the use of mathematical models of software systems for their specification and validation. Topics include finite state machine models, models of concurrent systems, verification of models, and limitations of these techniques.

Instructor(s)

Ardis

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

CSSE 230 - Fundamentals of Software Development III.

Textbook(s)

The Way of Z: Practical Programming with Formal Methods by Jonathan Jacky, Cambridge University Press, 1997, ISBN 0-521-55976-6.

Course Objectives

After successful completion of this course, students will:

- demonstrate formal correctness of simple procedures
- construct formal models of sequential software systems
- construct formal models of concurrent software systems
- verify attributes of formal models
- comprehend the costs and benefits of formal methods.

Course Topics

Formal specification in Z
Program verification
Formal specification and analysis in Petri nets
UML notations: activity diagrams, statecharts, sequence diagrams
Role of formal methods in software engineering

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, e, f, and k.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics and contains significant design.

Prepared by

Mark A. Ardis, 3/23/06; additional data added by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 374 Software Architecture and Design 1

Catalog Description

Introduction to the architecture and design of complete software systems, building on components and patterns. Topics include architectural principles and alternatives, design documentation, and relationships between levels of abstraction.

Instructor(s)

Steve Chenoweth

Required/Elective

Required.

Class/Laboratory Schedule

- Approx three 50 minute lecture / discussions per week
- One 50-minute in-class project "mini lab cycle" per week (where students play roles of architect, customer and developer)
- Two hours of joint design sessions, acting as design consultants with a CSSE 230 class (the prerequisite to this class)
- Four architecture document review workshops (50 minutes)
- During last 2 weeks, the course becomes primarily lab time – Approx 7.5 hours, with team project presentation at end.
- Lab work done primarily outside class, average from 2004 class was approx 50 hours per student total.

Prerequisites

CSSE 230 – Fundamentals of Software Development III

Textbook(s)

1. *Software Architecture in Practice*, 2/E, by Len Bass, Paul Clements, and Rick Kazman, all of the Software Engineering Institute. ISBN: 0-321-15495-9. Publisher: Addison Wesley Professional. Copyright: 2003. Format: Cloth; 560 pp. Published: 04/09/2003. Required.

2. *Software Design* (2nd Edition), by David Budgen. Addison Wesley; 2 edition (May 15, 2003), ISBN: 0201722194. Required.

Selected papers in software architecture (10 used in 2004).

Course Objectives

After successful completion of this course, students will:

- Work effectively with a team of software project stakeholders – including members of the development team, in software architecture and design activities.
- Use problem statements as a basis for design alternatives; deal with interactions among problems and solutions.
- Understand key decision-making activities necessary when architecting and partitioning software; recognize issues in managing architectural attributes and evaluate designs for feasibility and soundness.
- Apply fundamental design methods and strategies such as styles, components and patterns, creating a software architecture document.

Course Topics

- The nature of the design process & General Purpose Problem Solving
- The software design process
- Refactoring
- Design in the software development process
- Design qualities , CRC cards, UML design
- Describing a design solution.
- Transferring design knowledge
- Architectural styles
- Design representations – black box & white box
- The rationale for method, Problem statements "101"
- Design processes and design strategies
- The Gang of Four design patterns
- Introduction – The Architecture Business Cycle & What is Software Architecture?
- Case Studies in Architecture
- Creating an Architecture & Brainstorming 101
- Achieving Qualities
- Designing the Architecture
- Intro to Client Relations workshop
- Lateral Thinking 101
- Documenting Software Architectures – How to write the spec
- Building frameworks.

Lab Topics – none.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d and e.

Program Outcomes

This course addresses SE Program Outcomes SE1, SE4 and SE6.

Professional Component

This course provides 4 credits of engineering topics & contains significant design.

Prepared by

Steve Chenoweth; additional data added by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 375 Software Construction and Evolution

Catalog Description

Issues, methods and techniques associated with constructing software. Topics include detailed design methods and notations, implementation tools, coding standards and styles, peer review techniques, and maintenance issues.

Instructor(s)

Don Bagert

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures each week

Prerequisites

CSSE 230, Fundamentals of Software Development III

Textbook(s)

- *Software Maintenance: Concepts and* Practice, Second Edition by Penny Grubb and Armstrong A. Takang, World Scientific Publishing, ISBN 981-238-426-X
- *Code Complete: A Practical Handbook of Software Construction*, Second Edition by Steve McConnell, Microsoft Press, ISBN 0-7356-1967-0

Course Objectives

After successful completion of this course, students will:

1. Explain the software maintenance process model
2. Explain how to plan for and transition to maintenance
3. Participate on a software maintenance team
4. Develop software user documentation
5. Explain how software professionals effectively employ various implementation methods and tools studied in earlier courses
6. Tune software to meet performance objectives
7. Explain the Laws of Software Evolution

Course Topics

- The Software Maintenance Process
- Maintenance Planning
- Transition Planning
- Software Maintenance Projects
- Effective Software Construction

- Software Performance Issues
- The Laws of Software Evolution

Lab Topics

none

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcomes SE1 and SE5

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Don Bagert; March 28, 2006; additional data added 6/9/06

Computer Science and Software Engineering	CSSE 376 Software Quality Assurance

Catalog Description

Theory and practice of determining whether a product conforms to its specification and intended use. Topics include software quality assurance methods, test plans and strategies, unit level and system level testing, software reliability, peer review methods, and configuration control responsibilities in quality assurance.

Instructor(s)

Ardis

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

CSSE 230 - Fundamentals of Software Development III

Textbook(s)

- *Rapid Testing* by Robert Culbertson, Chris Brown, and Gary Cobb, Prentice Hall 2002, ISBN 0130912948.
- *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests* by Jeffrey Rubin, John Wiley and Sons 1994, ISBN 0471594032.

Course Objectives

After successful completion of this course, students will:

- Perform formal reviews of software artifacts
- Create a test plan for a software system
- Apply different strategies for unit-level and system-level testing
- Understand principles and strategies of integration and regression testing
- Use a problem tracking system
- Evaluate a user interface for suitability
- Understand purposes of quality processes, methods for measuring that quality, and standards used

Course Topics

Test plans and test case creation
Unit testing
System testing, including regression and acceptance testing
Usability testing

Reviews
Reliability

ABET Criteria

This course primarily addresses ABET Criteria a, b, g, and k.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Mark A. Ardis, 3/23/06; additional data added by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 377 Software Architecture and Design 2

Catalog Description

This is a second course in the architecture and design of complete software systems, building on components and patterns. Topics include architectural principles and alternatives, design documentation, and relationships between levels of abstraction.

Instructor(s)

Steve Chenoweth

Required/Elective

Required.

Class/Laboratory Schedule

- Approx three 50 minute lecture / discussions per week
- One 50-minute in-class project session per week
- Guest software architect speakers (2) for 3 x 50 minutes total
- Team presentation of architecture studies (4 x 50 minutes each)
- During last 2 weeks, the course becomes primarily lab time – Approx 7.5 hours, with team project presentation at end.
- Lab work done primarily outside class, average from 2004 class was approx 50 hours per student total.

Prerequisites

CSSE 230 – Fundamentals of Software Development III

Textbook(s)

1. *Software Architecture in Practice*, 2/E, by Len Bass, Paul Clements, and Rick Kazman, all of the Software Engineering Institute. ISBN: 0-321-15495-9. Publisher: Addison Wesley Professional. Copyright: 2003. Format: Cloth; 560 pp. Published: 04/09/2003. Required.
2. *Software Design* (2nd Edition), by David Budgen. Addison Wesley; 2 edition (May 15, 2003), ISBN: 0201722194. Required.
3. *Interaction Design*, by Jennifer Preece, Yvonne Rogers, and Helen Sharp. Wiley; 1 edition (January 17, 2002). ISBN: 0471492787. Required.

Selected papers in software architecture (6 used in 2004).

Course Objectives

After successful completion of this course, students will:

- Design and build effective human-computer interfaces using standard

methods and criteria. (This is something we don't do in depth, in any other course.)
- Understand the basic ingredients of successful software product lines – How to do multiple releases of software.
- Analyze the quality and economics of existing designs and systems – The heart of software architecture. Includes make vs. buy decisions, and discussion of component selection.
- Using UML and other methodologies and notations, create the overall design for a system and document this design – Elaborates on the ways to develop, prototype and document architectures.
- Practice the process by which architectures get created, in terms of technologies, economics, people and processes – Extends the project work of CSSE 374, looking at more patterns and new angles, including also some full-blown design methods.

Course Topics

- Understanding interaction basics
- Understanding users; How interfaces affect users
- Designing for collaboration
- Process of interaction design; Interaction Design (ID) requirements
- Design, prototyping and construction; User-centered design
- Patterns (extension from CSSE 374)
- Stepwise refinement, incremental design
- Designing with objects; Component-based design
- Reconstructing architectures
- ATAM - Analyzing architectures (Bass's terminology)
- CBAM – Financial analysis (Bass's terminology)
- Software product lines; Off-the shelf components
- Software architecture of the future

Lab Topics

None.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a, c, d and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits of engineering topics & contains significant design.

Prepared by

Steve Chenoweth; additional data added by Don Bagert 6/9/06

Computer Science and Software Engineering	CSSE 404 Compiler Construction

Catalog Description

Theory and practice of programming language translation. Lexical analysis, syntax analysis, parser generators, abstract syntax, symbol tables, semantic analysis, intermediate languages, code generation, code optimization, run-time storage management, error handling. Students will construct a complete compiler for a small language.

Instructor(s)

Clifton, Curt

Required/Elective

Elective

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

CSSE 232, Computer Architecture I
CSSE 304, Programming Language Concepts
CSSE/MA 474, Theory of Computation

Textbook(s)

Engineering a Compiler by Cooper and Torczon. Morgan Kaufmann, 2003.

Course Objectives

After successful completion of this course, students will be able to:

- implement a program to tokenize an input stream based on a regular language definition
- convert a left-recursive context-free grammar into a right-recursive one
- construct the canonical set of LR(1) items and derive the tables used by an LR(1) parser for a given context-free grammar
- implement, using a parser-generator such as ANTLR and CUP, a program to parse a token stream and construct an intermediate representation based on a context-free grammar
- implement code to perform context-sensitive analysis and type-checking on a program encoded in some intermediate representation
- implement code to construct symbol tables for a simple, imperative programming language
- implement a program that takes code written in a high-level language and generates corresponding code in a low-level target language, such as MIPS or Java Bytecode
- clearly explain selected, more advanced programming language

implementation issues (such as optimization or adding non-main-stream features), based on independent study, and extend a working compiler to address them

Course Topics

Regular Expressions
Nondeterministic Finite Automata
Deterministic Finite Automata
Scanner Generators
Context-free grammars
Backus-Naur Form
Top-Down Parsing (Eliminating Left-Recursion, Eliminating arbitrary look-ahead, First and Follow sets, Left-factoring)
Bottom-Up Parsing, Building LR(1) Tables
Parser Generators
Context-Sensitive Analysis (Attribute Grammars, Ad-hoc Syntax-Directed Translation)
Graphical Intermediate Representations (Abstract Syntax Trees, Directed Acyclic Graphs, Control Flow Graphs, Dependence Graphs)
Linear Intermediate Representations (Stack-machine code, Three-address Code, Static Single-Assignment Form)
Symbol tables
Type Systems and Practical Typechecking
Procedure abstraction
Code shape
Instruction scheduling (Tree-pattern matching, peephole optimization)
Redundancy elimination

Lab Topics

None.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Cary Laxer and Curt Clifton, 6/6/06; additional data added by Don Bagert 06/09/06

Computer Science and Software Engineering	CSSE 413 Artificial Intelligence

Catalog Description

Students investigate how to model and implement intelligent behavior using computers. Topics are chosen from how machines can: solve problems; reason and use knowledge; learn from experience; and perceive and act. Students explore these topics by implementing many of the ideas in software. Readings are drawn both from a textbook and from technical papers in recent conferences and journals.

Instructor(s)

Steve Chenoweth

Required/Elective

Elective

Class/Laboratory Schedule

- Four 50 minute lecture / discussions per week
- During last 2 weeks, this becomes primarily lab time – Approx 7.5 hours.
- Lab work done primarily outside class, average from 2004 class was approx 60 hours per student total.

Prerequisites

CSSE 220 – Fundamentals of Software Development II

Textbook(s)

Artificial Intelligence - A Modern Approach, by Stuart Russell and Peter Norvig, **Second Edition.** Prentice Hall, 2003. **required**

Course Objectives

After successful completion of this course, students will:

- Describe the fundamental design alternatives for AI and for intelligent agents.
- Use common tools for AI building systems.
- Develop a program to do heuristic search in a significant domain such as the TSP.
- Build a knowledge representation model and use it for decision making.
- Create an AI program for which quality results and timeliness are both critical.
- Apply the concepts of learning and probability to AI.
- Understand the philosophical and ethical issues surrounding AI.

Course Topics

- Intelligent Agents
- Search Strategies, Heuristic Search & Local Search
- Constraint Satisfaction Problems
- Adversarial Search & Games
- Logical Agents
- First-order Logic, Inference, Chaining and Resolution
- Knowledge Representation, Reasoning Systems
- Planning Problems
- Acting Under Uncertainty and Learning from Observations
- Learning Decision Trees, Knowledge in Learning
- Inductive Logic
- Statistical Learning Methods
- Neural Networks
- Philosophical Foundations
- Ethics and Risks of AI
- AI: Present and Future, Discussion of Impact of AI

Lab Topics

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Steve Chenoweth; additional data added by Don Bagert 06/09/06

Computer Science and Software Engineering	CSSE 432 Computer Networks

Catalog Description

Organization, design, and implementation of computer networks, especially the Internet. Network protocols, protocol layering, flow control, congestion control, error control, packet organization, routing, gateways, connection establishment and maintenance, machine and domain naming, security. Each of the top four layers of the Internet protocol stack: application (FTP, HTTP, SMTP), transport (TCP, UDP), network (IP), link (Ethernet).

Instructor(s)

Ardis

Required/Elective

Elective.

Class/Laboratory Schedule

Three 50-minute lectures and one 50-minute laboratory per week.

Prerequisites

CSSE 220 - Fundamentals of Software Development II.

Textbook(s)

Computer Networking: A Top-Down Approach Featuring the Internet (third edition) by James F. Kurose and Keith W. Ross, Addison Wesley, 2002, ISBN 0-321-22735-2.

Course Objectives

After successful completion of this course, students will:

- Understand vocabulary of computer networking
- Understand networking services and how they are provided by multiple layers of protocols
- Understand principles and mechanisms of congestion control in networks.
- Understand principles and mechanisms of routing in networks.
- Understand security mechanisms for networks and why they are needed.
- Be able to implement reliable file transfer services on an unreliable network.

Course Topics

Application layer of protocol stack: FTP, email, HTTP
Transport layer of protocol stack: UDP, TCP
Network layer of protocol stack: routing
Link layer of protocol stack: ethernet
Wireless networks

Network security
Network administration

Lab Topics

Protocol frame formats
Sliding window protocols
Error detection
File transfer protocol

ABET Criteria

This course primarily addresses ABET Criteria a, f, and k.

Program Outcomes

This course primarily addresses SE Program Outcomes 4 and 5.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Mark A. Ardis, 3/23/06; modified by Don Bagert 6/9/06

CSSE 442
Computer Security

Catalog Description

This course introduces ethical, theoretical, and practical issues of information security in computing systems. Implications of relevant professional codes of ethics are a recurring theme of the course. Foundational topics include access control matrices and standard system models, as well as policies for security, confidentiality, and integrity. Implementation issues include key management, cipher techniques, authentication, principles of secure design, representation of identity, access control mechanisms, information flow, life cycle issues, and formal evaluation and certification techniques. Additional topics include malicious logic, vulnerability analysis, and auditing. Computer network attack techniques are discussed and explored in a closed environment to motivate and inform discussion and exploration of computer network defense techniques.

Instructor(s)

Merkle

Required/Elective

Elective.

Class/Laboratory Schedule

Four 50-minute meetings per week, split roughly equally between lecture and in-class exercises.

Prerequisites

CSSE 332 and MA 275

Textbook(s)

Computer Security: Art and Science, Matt Bishop, Addison-Wesley, 2003.

Course Objectives

Students who successfully complete this course should be able to:

- Apply ethical theory to cyberethics issues.
- Apply the Access Control Matrix model to represent protection systems.
- Explain the principles that guide the design of authentication mechanisms.
- Apply rudimentary mathematical, statistical, and protocol-based cryptosystem attacks.
- Analyze and apply basic techniques and protocols for cryptographic key management, cryptographic key exchange, public key interchange, and digital signature
- Analyze and apply cryptographic protocols in practical environments;
- Explain the eight basic design principles for security mechanisms;

- Apply basic access control mechanisms
- Analyze information flows in a system to determine whether or not they conform to policy.
- Discuss current events, practices, and tools relevant to computer security.

Course Topics

- Cyberethics
- Security Law
- Access Control Matrices
- Authentication
- Security Policies
- Applications of Cryptography in Security
- Design Principles
- Access Control Mechanisms
- Information Flow
- Practical Aspects of Computer Network Attack and Defense

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Larry Merkle, October 9, 2005; additional data added by Don Bagert, June 9, 2006

Computer Science and Software Engineering	CSSE 451 Advanced Computer Graphics

Catalog Description

Advanced topics in computer graphics. Topics will be drawn from current graphics research and will vary, but generally will include ray tracing, radiosity, physically-based modeling, animation, and stereoscopic viewing. Programming assignments and a research project are required.

Instructor(s)

Laxer, Mellor

Required/Elective

Elective

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

CSSE 351, Computer Graphics

Textbook(s)

Computer Graphics with OpenGL (Third Edition) by Hearn and Baker. Pearson Prentice Hall, 2004.

Course Objectives

After successful completion of this course, students will be able to:

- Implement a ray tracer.
- Read, understand, and discuss current professional journal articles in computer graphics.
- Research and learn a graphics API (other than OpenGL) and develop software that generates realistic images using that graphics API.

Course Topics

Ray tracing
NURBS
Radiosity
Graphics APIs

Lab Topics

None

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Cary Laxer, 3/28/06; additional data added by Don Bagert 06/09/06

Computer Science and Software Engineering	CSSE461 Computer Vision

Catalog Description

An introduction to 3D computer vision techniques. Both theory and practical applications will be covered. Major topics include image features, camera calibration, stereopsis, motion, shape from x, and recognition.

Instructor(s)

Mellor

Required/Elective

Elective.

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

CSSE 220 – Fundamentals of Software Development II
MA 221 – Differential Equations and Matrix Algebra I

Textbook(s)

Multiple View Geometry in Computer Vision, Richard Hartley and Andrew Zisserman, 2003.

Course Objectives

After successful completion of this course, students will:

1. Understand the properties of projective lines, projective planes, and projective spaces.
2. Be able to use homographies to transform image data (e.g. remove perspective distortion).
3. Understand the pin-hole camera model and projection matrices.
4. Understand epipolar geometry.
5. Be able to calibrate a camera.
6. Be able to reconstruct a 3D scene from image data.

Course Topics

Image features
Projective geometry
Camera calibration
Stereopsis
Motion
Shape from *x*

Lab Topics

none

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Matthew Boutell, Course coordinator, 3/29/2006; additional data added by Don Bagert 06/09/06

Computer Science and Software Engineering	CSSE 481 Web-based Information Systems

Catalog Description

In this course, students learn about several aspects of research: thinking creatively about interesting research problems, researching existing work in a chosen area, and keeping current in a field. Students are exposed to the process of research by writing a pre-proposal for a project that advances the web. Projects either develop new web-technologies or applications or investigate a topic of importance. Based on feedback received, groups of students write a research proposal which goes through a formal peer review process. Approved projects are pursued for the remainder of the quarter. Students present current research as well as give a final presentation of their group project. Selected web-technologies are introduced; in the past, these have included CGI programming and XML technologies.

Instructor(s)

Wollowski

Required/Elective

Elective.

Class/Laboratory Schedule

Four 50 minute sessions each week.

Prerequisites

CSSE 230

Textbook(s)

None

Course Objectives

After successful completion of this course, students will be able to:

1. Stay current in the area of the world-wide web.
2. Identify and propose applications which advance the web.
3. Develop applications and perform research which advances the web.
4. Present their own work in a scientific manner.

Course Topics

- Identifying worthwhile projects
- Background research on current work
- Researching a project which advances the web
- Select current web-technologies
- Current and future developments of the web

Lab Topics

None

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcomes a and e.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits under engineering topics.

Prepared by

Michael Wollowski, 06/08/06; additional data added by Don Bagert 06/09/06

Computer Science and Software Engineering	CSSE 497-498-499 Senior Project I, II and III

Catalog Description

Group software engineering project requiring completion of a software system for an approved client. Tasks include project planning, risk analysis, use of standards, prototyping, configuration management, quality assurance, project reviews and reports, team management and organization, copyright, liability, and handling project failure. **Note that this is a three-term, three-course senior project sequence.** It is an integral part of the Computer Science and Software Engineering majors at Rose-Hulman Institute of Technology. By working with companies on real problems, students learn key aspects of software development, including risk analysis and quality assurance.

Instructor(s)

Ardis, Chenoweth (2005-6)

Required/Elective

Required for SE major. Either this or Senior Thesis is required for CS major.

Class/Laboratory Schedule

One 50-minute meeting with instructor each week. Additional team meetings and work by individuals and subgroups as required each week for the project. Twice per term, presentations are done by each team. Each team also meets with their client, potential users (if any) and other stakeholders as needed.

Prerequisites

CSSE 371 (Requirements Engineering) and 372 (Project Management).

Textbook(s)

None.

Course Objectives

This course exercises many skills acquired in earlier courses. Students who successfully complete this course will be able to demonstrate their ability to:

Communication Skills

- **Reading:** read technical documents and offer constructive criticism of their content and style
- **Writing:** write several different types of technical documents
- **Oral presentation:** prepare and deliver technical material at the appropriate level of detail

Management Skills

- **Leadership:** lead a small software team (if team leader) or ability to support the leadership of the team (if not)
- **Time management:** estimate and monitor personal time across multiple tasks
- **Meeting facilitation:** lead and participate in small groups in constructive meetings
- **Estimating:** estimate effort required to complete technical tasks
- **Risk:** assess project risks and plan mitigation strategies
- **Planning:** prepare a feasible plan for the accomplishment of several technical tasks

- **Monitoring:** track the progress of several tasks according to a plan

Technical Skills

- **Analysis:** analyze technical requirements and proposals for feasibility and to model the consequences of proposed solutions
- **Design:** construct appropriate abstractions of problems and solutions
- **Coding:** produce and inspect implementations of software according to project standards
- **Testing:** prepare test plans and to participate in both unit-level and system-level testing activities

Professionalism

- **Ethics:** identify and prevent unethical professional behavior
- **Intellectual property issues:** make appropriate professional judgments regarding choice of methods for protecting intellectual property
- **Social issues:** evaluate and avoid possible negative social aspects of a software product
- **Relationships with clients:** interact with clients in a professional manner

Course Topics

In general the course topics depend upon the actual project being pursued by a team. The exception is that certain deliverables are expected and so there is learning or reinforcement of prior learning on how to do these, as needed. The ultimate criterion for some of those is the preferences client, they generally include making formal project presentations and delivery of some or all of the following project artifacts:

- Project Plan
- Configuration Management Plan
- Individual Weekly Reports
- Team Weekly Reports
- Meeting Agendas and Minutes
- Requirements Specification
- Design Document
- Test Plans, Test Case Specifications and Test Results
- Software Manual

Lab Topics

The majority of the above course topics are done in a lab or meeting room environment, as appropriate.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a, b, c, d, e, f and g.

Program Outcomes

This course addresses SE Program Outcomes SE1, SE4, SE5, SE6, SE7 and SE8, and Institutional Outcomes RH8 and RH10.

Professional Component

This three-course sequence provides a total of 12 credits under engineering topics and contains significant design.

Prepared by

Steve Chenoweth, 6/5/06; additional data added by Don Bagert 6/9/06

Electrical and Computer Engineering ECE130
Introduction to Digital Logic

Catalog Description
Combinational logic design, Boolean algebra, logic minimization, Karnaugh maps, static and dynamic hazards, multiplexers and memories in combinational design, flip-flops, registers and counters, finite state machine design. Use of logic simulator for several design problems.

Instructor(s):
Doering, Mu, Song

Required/Elective:
Required

Class/Laboratory Schedule
Four 50-minute lectures per week; 20% of class days devoted to mini-projects.

Prerequisites:
None

Textbook(s):
Marcovitz, A.B., *Introduction to Logic Design*, 2nd ed. McGraw Hill, 2005.

Course Objectives:
After successfully completing this course the student should be able to:
- Appreciate the significance of digital information processing systems:
- Use binary representations of information:
- Analyze and design combinational circuits:
- Analyze and design sequential circuits:
- Analyze and design simple register-transfer level systems:
- Realize a circuit description as an FGPA-based system:

Course Topics
- Binary numbers and arithmetic; 2's complement signed numbers
- Boolean algebra, logic gates, truth tables, Karnaugh maps
- Switch-level circuits, propagation delay, hazards and glitches
- MSI components (adder, subtractor, MUX, encoder, decoder, comparator)
- ROMs, PLAs, FPGAs
- Latches and flip-flops, timing diagrams
- Sequential circuit analysis, state tables, state diagrams
- Register, shift register, synchronous counter
- Finite state machine design, manual timing analysis
- Datapath design, controller/datapath design

ABET Criteria

This course primarily addresses ABET Criteria a, b, e, g, k

Program Outcomes

This course primarily addresses SE Program Outcome SE1.

Professional Component

This course provides 3 credits of engineering science and 1 credit of engineering design.

Prepared by

Ed Doering, 5/11/06; modified by Don Bagert 6/9/06

Mathematics	**MA 111** **Calculus I**

Catalog Description

Calculus and analytic geometry in the plane. Algebraic and transcendental functions. Limits and continuity. Differentiation, geometric and physical interpretations of the derivative, Newton's method. Introduction to integration and the Fundamental Theorem of Calculus.

Instructor(s)

Various Mathematics instructors

Required/Elective

Required.

Class/Laboratory Schedule

Five (5) - 50 minute sessions per week

Prerequisites

It is assumed that the student has a mastery of high school algebra, pre-calculus and trigonometry concepts.

Textbook(s) and other required material

Textbook: *Thomas' Calculus - Early Transcendentals* , 11th Edition, Maurice Weir, Joel Hass, Frank Giordano 2006.
Computer Usage: Maple

Course Objectives

- Introduce students to differential calculus and beginning integration, including anti-derivatives and the Fundamental Theorem of Calculus;
- Introduce students to the application of differential calculus and beginning integration in science and engineering
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems.
- Develop student ability to communicate mathematically.

Course Topics

- Functions and Pre-Calculus review
    - Graph of a function y=f(x), domain/range.
    - Properties of functions and graphs, e.g., increasing/decreasing intervals, local max/min.
    - Definition and properties of polynomial, trigonometric, exponential and logarithmic functions, and relevant inverse functions.

- Parametric equations: physical interpretation as motion of a body

- Limits and Continuity
  - Limits.
  - Continuity
  - Intermediate value theorem
  -
- Differentiation
  - Average rate of change.
  - Instantaneous rate of change and definition of the derivative.
  - Formulas for elementary derivatives (polynomials, powers of x, $\sin(x)$, $\cos(x)$, $\tan(x)$, $\sec(x)$, e^x, $\ln(x)$, $\arcsin(x)$, $\arctan(x)$
  - Linearity, product, quotient, and chain rules.
  - Implicit differentiation, parametric curves.
  - Approximation by secant and tangent lines, differentials
  - Mean value theorem, Rolle's Theorem , Extreme Value Theorem

- Applications of Derivatives
  - limits at infinity
  - Graphical/physical interpretation of first and second derivatives
  - Derivatives as velocity and acceleration, motion problems (including motion described by parametric equations.)
  - Optimization problems.
  - Related-rate problems.
  - Newton's method

- Integration
  - Position from velocity, area under a curve
  - Riemann sums
  - Fundamental Theorem, anti-derivatives and properties, specifically
  - linearity, polynomials, powers of x, $1/x$, $\sin(x)$, $\cos(x)$, $\sec^2(x)$,
  - e^x, $1/\text{sqrt}(1-x^2)$, $1/(1+x^2)$.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a.

Program Outcomes

This course addresses Software Engineering Program Outcomes SE1, plus Institutional Outcomes RH7 and RH8.

Professional Component

This course provides 5 credits under mathematics and basic sciences

Prepared by

S. Allen Broughton 3/8/06; additional data inserted by Don Bagert 6/9/06

Mathematics	MA 112 Calculus II

Catalog Description

Techniques of integration, numerical integration, applications of integration. L'Hopital's rule and improper integrals. Separable first order differential equations, applications of separable first order differential equation. Series of constants, power series, Taylor polynomials, Taylor and McLaurin series.

Instructor(s)

Various mathematics instructors

Required/Elective

Required.

Class/Laboratory Schedule

Five 50-minute lectures per week.

Prerequisites

MA 111 - Calculus I or MA 102 - Differential Calculus or equivalent.

Textbook(s) and other required material

Textbook: *Thomas' Calculus - Early Transcendentals* , 11th Edition, Maurice Weir, Joel Hass, Frank Giordano 2006.
Computer Usage: Maple

Course Objectives

- Introduce students to integral calculus (including elementary first order differential equations.
- Introduce students to the application of the integral calculus and differential equations in science and engineering
- Introduction students to series of constants and functions, and the notions of approximation and convergence
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems
- Develop student ability to communicate mathematically.

Course Topics

- Integration---Basic Theory and Techniques
  - Riemann sums (review for Fall quarter freshman as needed)
  - Anti-derivatives for x^n , e^x, $1/x$, $\sin(x)$, $\cos(x)$, $\sec^2(x)$, $1/(x^2+1)$, 1/sqrt(1-x^2), $\cosh(x)$, and $\sinh(x)$
  - Linearity of integration

    - Integration by substitution
    - Integration by parts
    - Integration by partial fraction decomposition
    - Other integration techniques
    - L'Hopital's rule (may be taught in Calc I, review as needed)
    - Improper integrals
    - Numerical approximation using Trapezoidal and Simpson's rule

- Applications of Integration
    - Area
    - Displacement and distance travelled
    - Volumes of revolution (disk/shells)
    - Arc length, surface area of revolution
    - Work from force, potential energy

- Differential Equations
    - Definition, order, linearity
    - Separation of variables for separable first order equations.
    - Application to exponential growth and decay, population growth (logistic equation), Newton's law of cooling, salt tank problems, falling bodies (with and without air resistance.)
- Series
    - Series of constants
    - convergence
    - ratio test, integral test
    - power series
    - Taylor polynomials
    - Taylor and Mclaurin series of basic functions

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a.

Program Outcomes

This course addresses Software Engineering Program Outcomes SE1, plus Institutional Outcomes RH7 and RH8.

Professional Component

This course provides 5 credits under mathematics and basic sciences

Prepared by

S. Allen Broughton 3/8/06; additional data inserted by Don Bagert 6/9/06

Mathematics MA 113
Calculus III

Catalog Description
Vectors and parametric equations in three dimensions. Functions of several variables, partial derivatives, maxima and minima of functions of several variables, multiple integrals, and other coordinate systems. Applications of partial derivatives and multiple integrals.

Instructor(s)
Various Mathematics instructors

Required/Elective
Required

Class/Laboratory Schedule
Five (5) - 50 minute sessions per week

Prerequisites
MA112 - Calculus II or equivalent.

Textbook(s) and other required material
Textbook: *Thomas' Calculus - Early Transcendentals,* 11th Edition, Maurice Weir, Joel Hass, Frank Giordano 2006.
Computer Usage: Maple

Course Objectives
- Introduce students to multivariable differential and integral calculus, and more vector techniques, especially in three dimensions
- Introduce students to the application of multivariable calculus and in science and engineering.
- Develop student mathematical modeling and problem solving skills.
- Develop student ability to use a computer algebra system (CAS) to aid in the analysis of quantitative problems.
- Develop student ability to communicate mathematically.

Course Topics
- Vectors in Three Dimensions, Vector-valued functions
    - Space coordinates and vectors in space
    - Dot product and projection (review from MA 111, maximum of one day)
    - Cross products
    - Lines and planes in space
    - Vector-valued functions
    - Differentiation/Integration of vector-valued functions

    - Curvature, arc length, unit tangent and normal vectors, components of acceleration

- Multivariable Differential Calculus
    - Functions of several variables
    - Partial derivatives
    - Chain rule
    - Directional derivatives, gradients
    - Tangent planes and normal lines
    - Unconstrained extrema
    - Lagrange multipliers

- Multivariable Integral Calculus
    - Double integrals, evaluation
    - Polar coordinates, change of coordinates
    - Triple integrals, evaluation
    - Cylindrical and spherical coordinates, integration in these coordinate systems

- Applications
    - Velocity and acceleration problems, projectile motion
    - Unconstrained and constrained multivariable max/min problems
    - Volume, surface area
    - Mass, moments, moment of inertia

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a.

Program Outcomes

This course addresses Software Engineering Program Outcomes SE1, plus Institutional Outcomes RH7 and RH8.

Professional Component

This course provides 5 credits under mathematics and basic sciences

Prepared by

S. Allen Broughton 3/8/06; additional data inserted by Don Bagert 6/9/06

Mathematics MA 221
Differential Equations and Matrix Algebra 1

Catalog Description
Basic matrix algebra with emphasis on understanding systems of linear equations from algebraic and geometric viewpoints, including the least squares process and eigenvalues and eigenvectors. First order differential equations including basic solution techniques and numerical methods. Second order linear, constant coefficient differential equations, including both the homogeneous and non-homogeneous cases. Introduction to complex arithmetic, as needed. Applications to problems in science and engineering.

Instructor(s)
Various mathematics instructors

Required/Elective
Required

Class/Laboratory Schedule
Four 50-minute lectures per week.

Prerequisites
MA 113 Calculus III or equivalent or permission of mathematics department head.

Textbook(s) and other required material
Textbook*: Differential Equations & Linear Algebra with Boundary Values Problems* - Edwards/Rose-Hulman, 2006
Computer Usage: Maple

Course Objectives
- Develop a deeper understanding of equations and their solutions, especially linear algebraic and differential equations.
- Improve mathematical modeling and analytical problem solving skills.
- Develop ability to communicate mathematically.
- Improve skill using the computer as a tool for mathematical analysis and problem solving.
- Introduce applications of mathematics, especially to science and engineering.

Course Topics
- **First order differential equations**
    - Review basic notions (e.g., separation of variables, initial value problems)
    - Review $dx/dt = ax$ and $dx/dt = ax + b$ and show structure of solution as particular solution plus homogeneous solution

- Numerical methods (e.g., Euler, RK4)
- Applications as appropriate

- **Second order linear differential equations**
  - Constant coefficient, homogeneous case (solving the characteristic equation requires basics of complex arithmetic through Euler's formula)
  - Method of undetermined coefficients for non-homogeneous case
  - Method of variation of parameters for non-homogeneous case
  - Resonance
  - Applications as appropriate
  -

- **Matrix Algebra**
  - Matrix Arithmetic (e.g., addition, scalar multiplication, matrix multiplication, inverses --emphasis on the 2 x 2 case for "by-hand" inverting)
  - Understanding a matrix A as a transformation (avoiding general vector space ideas) ; $Ax = b$
  - Representation of systems of linear equations as matrix equations
  - Gaussian Elimination for solving $Ax = b$.
  - Structure of general solution for $Ax = b$: algebraic point of view (particular solution plus homogeneous solution);
  - Understanding the Least Squares Process: algebraic point of view (solve the normal equations); geometric point of view (project b onto the range of A and solve)
  - Eigenvalues and Eigenvectors: algebraic point of view (solve $Ax = \lambda x$ for x and λ), and geometric point of view (Ax is parallel to x)
  - Applications as appropriate.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a.

Program Outcomes

This course addresses Software Engineering Program Outcomes SE1, plus Institutional Outcomes RH7 and RH8.

Professional Component

This course provides 4 credits of under mathematics and basic sciences

Prepared by

S. Allen Broughton 3/8/06; additional data inserted by Don Bagert 6/9/06

Mathematics **MA 275**
Discrete and Combinatorial Algebra I

Catalog Description

An introduction to enumeration and discrete structures. Permutations, combinations and the pigeonhole principle. Elementary mathematical logic and proof techniques, including mathematical induction. Properties of the integers. Set theory. Introduction to functions.

Instructor(s)

Various mathematics Instructors

Required/Elective

Required for CS, SE, CPE

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

No Rose-Hulman course, but is assumed that the student has a mastery of high school algebra, pre-calculus and trigonometry concepts.

Textbook(s) and other required material

Textbook: *Discrete and Combinatorial Mathematics, an Applied Introduction* 5th edition - Ralph P. Grimaldi

Course Objectives

- Introduce student to discrete mathematics covering especially the topics noted below.
- Improve mathematical modeling and analytical problem solving skills, especially with discrete mathematics topics.
- Develop ability to communicate mathematically.
- Introduce applications of discrete mathematics, especially to science and engineering.

Course Topics

- Logic
- Set Theory
- Proof Techniques
    - Induction
    - Proof by contradiction
    - Proof by contrapositive
- Counting Techniques
    - Permutations

    - Combinations
    - Pigeonhole principle
- Properties of integers
- Functions

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a.

Program Outcomes

This course addresses Software Engineering Program Outcomes SE1, plus Institutional Outcomes RH7 and RH8.

Professional Component

This course provides 4 credits under mathematics and basic sciences

Prepared by

S. Allen Broughton 3/8/06; additional data inserted by Don Bagert 6/9/06

Mathematics **MA 375**
Discrete and Combinatorial Algebra II

Catalog Description

An introduction to enumeration and discrete structures. Permutations, combinations and the pigeonhole principle. Elementary mathematical logic and proof techniques, including mathematical induction. Properties of the integers. Set theory. Introduction to functions.

Instructor(s)

Various mathematics Instructors

Required/Elective

Required for CS, SE

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

MA275 – Discrete and Combinatorial Algebra I

Textbook(s) and other required material

Textbook: *Discrete and Combinatorial Mathematics, an Applied Introduction* 5th edition - Ralph P. Grimaldi

Course Objectives

- Continue to introduce students to discrete mathematics, covering especially the topics noted in the course topics below.
- Improve mathematical modeling and analytical problem solving skills, especially with generating functions and recurrence relations.
- Develop an ability to communicate mathematically.
- Introduce applications of discrete mathematics, especially in science and engineering.

Course Topics

- Languages and Finite State Machines
  - Minimization of finite state machines
- Relations
  - Equivalence relations
  - Partial orders and Hasse diagrams
  - Relation matrices
  - Modular arithmetic

- Principle of Inclusion-Exclusion

- Generating functions ordinary and exponential
- Recurrence relations

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a.

Program Outcomes

This course addresses Software Engineering Program Outcomes SE1, plus Institutional Outcomes RH7 and RH8.

Professional Component

This course provides 4 credits under mathematics and basic sciences

Prepared by

S. Allen Broughton 3/8/06; additional data inserted by Don Bagert 6/9/06

Mathematics **MA 381**

Introduction to Probability with Applications to Statistics

Catalog Description

Introduction to probability theory; axioms of probability, sample spaces, and probability laws (including conditional probabilities). Univariate random variables (discrete and continuous) and their expectations including these distributions: binomial, Poisson, geometric, uniform, exponential, and normal. Introduction to moment generating functions. Introduction to jointly distributed random variables. Univariate and joint transformations of random variables. The distribution of linear combinations of random variables and an introduction to the Central Limit Theorem. Applications of probability to statistics.

Instructor(s)

Various Mathematics instructors

Required/Elective

Required for CS, SE, EE, CPE, replaces MA223

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

MA 113 Calculus III

Textbook(s) and other required material

Fundamentals of Probability with Stochastic Processes, 3rd Edition, by Saeed Ghahramani.

Course Objectives

- An understanding of the ideas of probability and probability modeling, including sample spaces, axioms, discrete and continuous random variables, univariate and joint distributions, moment generating functions, the central limit theorem, and basic statistical inference
- An understanding of the special language, notation, and point of view of probability
- An understanding of the concepts of probability necessary to undertake basic modeling and decision making in math, science, and engineering
- The ability to solve standard computational problems in probability, which includes using the computer as a tool for mathematical analysis and problem solving
- An understanding of the relationship between random variables and their distribution functions
- The ability to recognize special models, such as Bernoulli trials or Poisson

processes
- An understanding of how probability is applied to inferential statistics
- The ability to communicate in mathematical terms
- Using simulation as a tool in approximating probabilities
- Preparing for the types of problems you may see if you take the first actuarial exam

Course Topics

- Axioms of Probability: Sample space, events, axioms of probability, basic theorems
- Combinatorial Methods: Basic Counting Principles, Combinations, Permutations
- Conditional Probability and Independence**:** Conditional probability, laws of multiplication and total probability, Baye's formula, independence
- Distribution Functions and Discrete Random Variables: Random variables, distribution functions, discrete random variables, expectations of discrete random variables, variances and moments of discrete random variables, standardized random variables.
- Special Discrete Distributions**:** Bernoulli and binomial, Poisson (including Poisson process), and geometric distributions
- Continuous Random Variables: Probability density functions, density function of a function of a random variable, expectations and variances
- Special Continuous Distributions: Uniform, normal, exponential distributions
- Chapter 8. Bivariate Distributions**:** Joint distributions, independent random variables, conditional distributions
- More expectations and variances: Expected values of sums of random variables, covariance, correlation, conditioning on random variables
- Sums of independent random variables: Moment generating functions, sums of independent random variables, Markov and Chebyshev inequalities, central limit theorem

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome a.

Program Outcomes

This course addresses Software Engineering Program Outcomes SE1, plus Institutional Outcomes RH7 and RH8.

Professional Component

This course provides 4 credits under mathematics and basic sciences

Prepared by

S. Allen Broughton 3/8/06; additional data inserted by Don Bagert 6/9/06

Physics and Optical Engineering	PH 111 Physics I

Catalog Description

Newton's laws of motion, gravitation, Coulomb's law, Lorentz force law, strong and weak nuclear forces, conservation of energy and momentum, torque and angular momentum, relevant laboratory experiments.

Instructor(s)

Sudipa Kirtley

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures per week and one 160-minute lab every other week

Prerequisites

None

Textbook(s)

Physics for Scientists and Engineers, R. Knight, Pearson-Addison & Wesley, 2004.

Course Objectives

After successful completion of this course students will:

Calculus-based introductory treatment of fundamental laws of mechanics with emphasis on development of problem-solving skills.

Course Topics

A. Vectors: Scalars and vectors, decomposition of vectors into components, addition of vectors, cross and dot products, unit vectors.

B. Kinematics: Position, displacement, average and instantaneous velocity and acceleration equations for vertical and horizontal motions, definition of period, angular velocity and acceleration, relation between linear and angular terms, centripetal and tangential accelerations.

C. Newton's laws of motion with application. Forces and Newton's Laws: Forces between two surfaces, Newton's three laws, calculation of resultant acceleration, calculations involving light strings and pulleys, coefficients of static and dynamic friction.

D. Gravitation: Newton's law of gravity, universal gravitational constant, how to find the mass of the sun and the period of earth's orbit.

E. Electric Field and Coulomb's Law: Electric charges, electric potential, equipotential lines, lines of force, force on a charge in an electric field.

F. Magnetism: Magnetic field, lines of force and equipotential lines for a magnetic field, Earth has a magnet, force on a moving charge in a magnetic field.

G. Momentum, Impulse, and Collisions: Definitions, Newton's second law in momentum form, law of conservation of overall momentum, elastic and inelastic collisions, one- and two-dimensional collisions.

H. Work: Work done by a force from a force-distance graph, definition of kinetic energy, and its relation to the work done by an accelerating mass. Potential energy, conservation of mechanical energy in the absence and presence of dissipatory forces.

Lab Topics

- Analysis of the motion of falling bodies, including acceleration due to gravity and terminal velocity. Sonic Ranger and PC computer used for data collection and analysis. (1 week)
- Conservation of momentum using an air-track. Sonic Ranger and PC computer used for data collection and analysis. (1 week)
- Newton's Second Law using falling mass connected to an air-track glider via a string draped over a pulley. Sonic Ranger and PC computer used in data collection and analysis. (1week)
- Analysis of pendulum period and maximum velocity as a function of amplitude of swing, shaft encoder and PC computer used in data collection and analysis. (1 week)

ABET Criteria

This course primarily addresses ABET criteria a, b, and g.

Program Outcomes

This course primarily addresses SE Program Outcome SE1 and RHIT Institutional Outcomes RH6, RH7, RH8 and RH9.

Professional Component

This course provides 4 hours in the mathematics and basic sciences category.

Prepared by

Sudipa Kirtley, 2005; additional data inserted by Don Bagert 6/9/06

Physics and Optical Engineering	PH 112 Physics II

Catalog Description

Oscillations, one-dimensional waves, introduction to quantum mechanics, electric fields and potentials, electric current and resistance, DC circuits, capacitance, relevant laboratory experiments.

Instructor(s)

Charles Joenathan

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures per week and one 160-minute lab every other week

Prerequisites

PH111, Physics I and MA 111; Co-requisite : MA 112

Textbook(s)

Physics for Scientists and Engineers, R. Knight, Pearson-Addison & Wesley, 2004.

Course Objectives

After successful completion of this course students will:

Calculus-based introductory treatment of fundamental laws of mechanics with emphasis on development of problem-solving skills.

Course Topics

A. Torque, moment of inertia, angular momentum as used in the analysis of rotational motion. Equations of motion in terms of angular variables, Linear and angular variables, Rotational kinetic energy and rotational inertia, Parallel-axis theorem, Torque, moment arm, line of action of a force, Newton's second law in angular form, Work and rotational kinetic energy, Rolling bodies, KE in terms of center of mass, Angular momentum of a system of particles, and of a rigid body, Conservation of angular momentum.

B. Oscillatory motion. Simple harmonic motion: frequency, period, amplitude, angular frequency, wave number, phase. Velocity and acceleration amplitudes, Linear oscillator, Energy, Pendulums

C. Vibrations and sound waves. Transverse and longitudinal waves, Equation of a traveling wave, Wave speed on a stretched string, Power, Superposition of waves, Interference of waves, Standing waves and resonance.

D. Coulomb's Law and its applications. Electric field and field lines, E due to a system of point charges and a continuous charge distribution, Force and field, Dipole, and dipole in the presence of E (torque, and PE)

E. Electric field Gauss's Law. Electric flux, Application of G's Law to find the E due to a line charge, a spherical charge and a cylindrical distribution of charge, E inside a conductor
F. Electric potential Electric potential energy, Work done by electric field, Finding V from E, Finding E from V, Finding V for a group of charges and a continuous distribution of charges
G. Capacitance and dielectrics Capacitors, and capacitance. Finding C for different capacitors, Capacitors in parallel and series, Potential energy stored in a capacitor, Capacitors and dielectrics
H. Current, resistance and electromotive force. Electric current, Current density, drift velocity, charge carriers, Resistance and resistivity, Ohm's Law, Power
I. Direct current circuits. Emf, Kirchhoff's Laws, Resistances in series and parallel, Multiple loop circuits, RC Circuits (charging and discharging a capacitor, behavior of V, q, and I as a function of time in the two cases)
J. Electromagnetic radiation. Origin of EM radiation and the broad form of the spectrum. Definition and examples of black body radiation, Wien's and Stefan's Laws, Planck's solution for blackbody radiation.
K. Electromagnetic waves as particles: Photoelectric Effect and Compton Scattering, Particles as Waves, interference, wavelength of a particle. Uncertainty Principle, brief description of tunneling, quantization of energy

Lab Topics

- Investigation of torques and moment of inertia through measurement of angular acceleration of a rotating plate
- An investigation of the characteristic resonance frequencies of a string under tension (held at both ends), and a tube (open at one end).
- Study of current-voltage relationship in a DC circuit. Application of Kirchhoff's laws.
- Study the charging of a capacitor in an RC-circuit using ADC and PC-compatible computer.

ABET Criteria

This course primarily addresses ABET criteria a, b, g.

Program Outcomes

This course addresses SE Program Outcome SE1 and RHIT Institutional Outcomes RH6, RH7, RH8 and RH9.

Professional Component

This course provides 4 hours in the mathematics and basic sciences category.

Prepared by

Charles Joenathan, 2005; additional data inserted by Don Bagert 6/9/06

Chemistry — CHEM 201
Engineering Chemistry I

Catalog Description
Topics include stoichiometry, nomenclature, reactions in aqueous solution, atomic structure and periodic properties. Chemical bonding including Lewis dot structures and molecular geometry are considered in relation to the properties of solids, liquids, solutions and gases. Chemical kinetics is covered.

Instructor(s)
DeVasher

Required/Elective
Required.

Class/Laboratory Schedule
Three 50-minute lectures and one 150-minute lab per week.

Prerequisites
MA 111 – Calculus I

Textbook(s)
Chemistry, 8th Edition, Chang, Raymond, McGraw-Hill, 2005.

Course Objectives
After successful completion of this course, students will:
- be able to name simple ionic compounds and simple inorganic molecules correctly using systematic nomenclature
- possess an understanding of reaction stoichiometry, and predict the quantity of products being formed in a chemical reaction given a balanced chemical equation and known amounts of starting materials
- be able to take simple, accurate scientific measurements of matter (density, mass, volume, electrical conductivity) and propagate uncertainty in those measurements
- predict products of metathesis and combustion reactions
- be able to illustrate the structure of simple inorganic molecules using Lewis Dot Theory and predict the geometry of simple inorganic molecules using Valence Shell Electron Pair Repulsion Theory
- posses and understanding of the relative rates of chemical reactions given experimental data

Course Topics
Chemical nomenclature of simple ionic compounds and simple inorganic molecules
Stoichiometry

Gas Laws and Phase Changes
Dipole moments and Intermolecular forces
Valence Shell Electron Pair Repulsion Theory
Aqueous Phase Chemistry (metathesis reactions, oxidation-reduction reactions, combustion reactions)
Chemical kinetics

Lab Topics

Weight Percentage and Density of a Solution
Thermal Decomposition of Sodium Hydrogen Carbonate
The Ideal Gas Law
Molecular Models
The Packing of Atoms
Solid State
Electrical Conductivity of Solutions
Qualitative Reduction-Oxidation (REDOX) Reactions
Determination of the Rate Constant & the Order of a Reaction

ABET Criteria

This course primarily addresses ABET Criteria 3 Outcome a.

Program Outcomes

This course addresses SE Program Outcome SE1.

Professional Component

This course provides 4 credits of engineering science, 0 credits of basic science, 0 credits of basic math, and 0 credit of engineering design.

This course is an introduction to chemistry as it applies to engineers.

Prepared by

Rebecca B. DeVasher, 03/17/06; additional data inserted by Don Bagert 06/09/06

Catalog Description

RH 131 Rhetoric and Composition 4R-0L-4C F,W,S
Examines selected pieces of writing which are used as models for student composition. Emphasizes the use of evidence and methods of argumentation. Required of all students. (May not be counted for an Area Minor in Language and Literature.)

Instructor(s)

Carlson, Carvill, Dyer, Minster, Smith

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minutes periods per week.

Prerequisites

None.

Textbook(s)

Varies by instructor.

Course Objectives

- to identify, evaluate, and construct effective arguments;
- to produce a clean, efficient, and grammatically correct style
- to adapt to a variety of rhetorical situations and audiences;
- to read critically;
- to conduct research and document sources

Course Topics

Rhetorical Strategies: Audience Analysis, Genre, Voice, etc.
Process: Prewriting, Drafting, Revising, Editing
Research Methods, Documentation, and Assessment of Evidence
Critical Reading

Lab Topics

None.

ABET Criteria

This course primarily addresses ABET Criterion 3 Outcome g.

Program Outcomes

This course addresses SE Program Outcome SE5.

Professional Component

Four hours of general education

Prepared by

Caroline Carvill, March 2006; additional data inserted by Don Bagert 6/9/06

Humanities and Social Sciences	RH 330 Technical Communication

Catalog Description

RH 330 Technical Communication 4R-0L-4C F,W,S Pre: RH 131 and Junior class standing or consent of instructor

Discusses the preparation and presentation of engineering reports, both oral and written. (May not be counted for an area minor in Language and Literature.)

Instructor(s)

House, Minster, Watt, Williams, and Zoetewey

Required/Elective

Required.

Class/Laboratory Schedule

Four 50-minute lectures per week.

Prerequisites

RH 131 – Rhetoric and Composition

Textbook(s)

Laura J. Gurak and John M. Lannon, *A Concise Guide to Technical Communication*. 2nd Edition. New York: Pearson Longman, 2004

Edward R. Tufte, *The Cognitive Style of PowerPoint*. Cheshire, CT: Graphics Press LLC, 2003

Course Objectives

After successful completion of this course, students will understand:

- writing in context, including analyzing professional cultures in technical fields and professions; analyzing social contexts and audiences to determine how they shape the various purposes and forms of writing; communicating the technical aspects of the student's field or discipline to non/specialists; and understanding the ethical implications of working within the nexus of technology and culture.
- writing process, including developing strategies for planning, researching, drafting, revising, and editing documents that respond effectively and ethically to technical situations and audiences and developing strategies for managing short and long term projects.
- collaboration, including learning and applying strategies for effective collaboration within and across teams with overlapping interests; developing effective models for distribution of labor; communicating orally, in writing, and electronically with colleagues within and across hierarchies; negotiating conflict; and responding constructively to peers' work.

- research, including field research (e.g. analyzing the needs of an audience and supplying appropriate technical information to meet their needs) and library research (e.g. locating non/technical information and properly documenting it in the student's own work).
- document design, including designing usable, clear, persuasive, and accessible documents that meet multiple user and reader needs and learning to interpret and argue with visual data.

Course Topics

Audience analysis; discourse community analysis; rhetoric and persuasion; ethics in communication; summary writing; memo writing; peer review strategies; team communication; intercultural communication; case study analysis; elements of page design/layout; effective use of visuals; fundamentals of effective writing (grammar, spelling, punctuation, mechanics).

Lab Topics

Not applicable.

ABET Criteria

This course primarily addresses ABET Criteria g.

Program Outcomes

This course addresses SE Program Outcome SE5 and Institutional Outcome RH 6.

Professional Component

Four hours of general education

Prepared by

Julia M. Williams, 5/16/06; additional data inserted by Don Bagert 6/9/06

College and Life Skills CLSK 100

Catalog Description

This course will assist the student in acquiring life skills & in learning more about themselves. These new skills will assist the student in a smooth transition from high school to college and will provide the students with the tools necessary for success as a student and in life. Additionally, this course will introduce students to people & resources at Rose-Hulman Institute of Technology who can assist them in providing a positive educational and personal experience.

Instructor(s)

Tom Miller, curriculum coordinator (9 instructors each academic year)

Required/Elective

Required.

Class/Laboratory Schedule

One 50-minute lecture per week fall quarter, 10 class meetings

Prerequisites

None

Textbook(s)

None

Course Objectives

- Assist a first-year student in a smooth transition from high school to college
- Course will introduce students to professional staff and resources on campus

Course Topics

- Introduction to campus life/Greek life
- Office of the Registrar
- Electronic portfolio (Rose E-portfolio)
- Introduction to the Learning Center/resources
- Introduction to the Logan Library and research resources
- Introduction to Career Services
- Workshop on resume writing and eRecruiting
- Professional practices and experiences in professional engineering (Rose-Hulman Ventures)
- Introduction to Alumni Affairs and Student Alumni Association (SAA)
- Overview and results of the Myers-Briggs Personality Inventory

Lab Topics

None

ABET Criteria

Program Outcomes

Professional Component

This course provides 1 hour of general education.

Prepared by

Tom Miller, 5/2/06; modified by Don Bagert 6/9/06

Appendix I

(continued)

C. Faculty Resumes

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Cary Laxer, Professor and Head of Computer Science and Software Engineering

Degrees:

Ph.D.	1980	Biomedical Engineering	Duke University
B.A.	1976	Computer Science and Mathematics	New York University

Rose-Hulman Service:

Years of Service at Rose-Hulman:	25 years
Original Appointment Date and Rank:	September, 1981, Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s):	September, 1984, Associate Professor
	September, 1993, Professor

Professional Experience:

1981 – Present	**Assistant Professor/Associate Professor/Professor (Department Head since 2002)** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN
1999 – 2000	**Visiting Professor** Department of Information Systems and Computing UNITEC Institute of Technology, Auckland, New Zealand
1989 – 1990	**Lilly Visiting Professor of Emerging Cardiovascular Technologies** Duke University-University of North Carolina NSF Engineering Research Center Duke University, Durham, NC

Consulting Activities: University of Virginia's College at Wise – consulted on development of new computer science and software engineering programs, 2003.
Harvey Mudd College – external evaluator of computer science program, 2006.

Patents and Disclosures: None.

Professional Registration: None.

Principal Publications of the last five years:

C. Laxer and A. Young. "Quality Assurance: How much is needed?" Proceedings of the 13th Annual Conference of the NACCQ, 179-182, 2000.

C. Laxer. "Using Computer Graphics to Reinforce Mathematics." Proceedings of the 17th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education, Short Papers and Works in Progress, 205-209, 2000.

C. Laxer. "Treating Computer Science as Science: An Experiment with Sorting." Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education, 189, 2001.

M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y.B. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz. "Report by the ITiCSE 2001 Working Group on Assessment of Programming Skills of First-year CS Students." SIGCSE Bulletin, 33:4, 125-140, 2001.

C. Laxer. "Evaluating Students Team Project Experiences." Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education, 198, 2002.

C. Laxer. "A Student Team Research Approach to the Second Course in Computer Graphics." Proceedings of CGE02 Eurographics - ACM SIGGRAPH Workshop on Computer Graphics Educations, 1-3, 2002.

M. Dick, J. Sheard, C. Bareiss, J. Carter, D. Joyce, T. Harding, and C. Laxer. "Addressing Student Cheating: Definitions and Solutions," SIGCSE Bulletin, 35:2, 172-184, 2003.

S. Alexander, J. Amillo, R. Boyle, M. Clark, M. Daniels, C. Laxer, K. Loose, and D. Shinners-Kenndy. "Case Studies in Admissions to and Early Performance in Computer Science Degrees," SIGCSE Bulletin, 35:4, 137-147, 2003.

S. Cunningham, W. Hansmann, C. Laxer, and J. Shi. "The Beginning Computer Graphics Course in Computer Science," Computer Graphics, 38:4, 24-25, 2004.

Membership in Scientific and Professional Societies:

ACM: SIGCSE, SIGGRAPH
IEEE (Senior Member): Computer Society
ASEE

Selected Honors and Awards:

Recipient of Dean's Outstanding Teacher Award, Rose-Hulman Institute of Technology, 1987

President's Outstanding Faculty Member of the Week, Rose-Hulman Institute of Technology, 1990, 1992 (twice), 1993 (twice), 1995, 1996 (three times), 1998, 1999, 2000, 2001, 2002 (twice), 2003, 2004 (twice)

ACM Recognition of Service Award, 1990, 1993 (twice), 1995, 1996, 2000, 2001

Member of Pi Mu Epsilon, Upsilon Pi Epsilon, and Blue Key honorary societies

Life Member of Alpha Phi Omega national service fraternity (Tau Lambda Chapter Distinguished Service Key, 1991; National Distinguished Alumnus Key, 1998)

Listed in Outstanding Young Men of America, Who's Who in Frontier Science and Technology, Who's Who Among America's Teachers, American Men and Women of Science

Member of European Academy of Sciences

Named Teacher of the Year by Triangle Fraternity, 1984, 2004

Honorary Alumnus Award, Rose-Hulman Alumni Association, 2002

Institutional and Professional Service of the last five years:

Registrar, ACM SIGCSE ITiCSE Conference, 2004, 2006.
Program Chair, AM SIGGRAPH/EUROGRAPHICS CGE04, 2004.
Registrar, ACM SIGCSE Technical Symposium, 1996-date.
Member of Faculty Affairs Committee 2000-2002; chair 2001-2002.
Member of Curriculum Committee, 2002-date.
Member of Fraternity Advisory Council, 1988-date.
Chapter Advisor to Triangle Fraternity, 1988-date.
Faculty advisor to approximately 20-25 students majoring in computer science, annually.

Professional Development Activities of the last five years:

Attended SIGCSE, ITiCSE, and SIGGRAPH conferences annually.
Reviewer for SIGCSE, ITiCSE, and CGEMS.

Co-chaired ACM SIGGRAPH Education Committee workshop on defining a computer graphics curriculum, 2003-date

Program chair for CGE04, Computer Graphics Educators Workshop, Zhejiang University, Hangzhou, China
Participant in CGE02, Computer Graphics Educators Workshop, University of Bristol, Bristol, England

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Claude W. Anderson, III, Professor of Computer Science and Software Engineering

Degrees:

Masters	1987	Computer Science	Indiana University (Bloomington)
Ph.D.	1981	Mathematics	University of Illinois (Urbana)
Masters	1977	Mathematics	University of Illinois (Urbana)
B.S.	1975	Mathematics	Caltech

Rose-Hulman Service:

Years of Service at Rose-Hulman:	18 years
Original Appointment Date and Rank:	September, 1988, Associate Professor
Date(s) of Promotion(s) and Resulting Rank(s):	September, 1994, Full Professor

Professional Experience:

1988 – Present	**Assoc. Professor/Professor** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN
2005 – 2006	**Software Engineer** ANGEL Learning, Inc., Indianapolis, IN
Summer 1998	**Software Engineer** Beckman Instruments, Indianapolis, IN
1996-1997	**Software Engineer** Beckman Instruments, Indianapolis, IN
Summers 1994-95	**Software Engineer** Navistar Technical Center, Fort Wayne, IN
1981-1988	**Assistant Professor, Mathematics and Computer Science** Wilkes College, Wilkes-Barre, PA

Consulting Activities: Software design and development, Pennsylvania Funeral Directors Association.

Patents and Disclosures: None.

Professional Registration: None

Principal Publications of the last five years:
"Transitioning to an Objects-Early Three-Course Introductory Sequence: Issues and Experiences" proceedings of the SIGCSE Computer Science Education Conference, Norfolk, VA, February, 2004.

Membership in Scientific and Professional Societies:
Association for Computing Machinery (ACM)

Selected Honors and Awards:
Carpenter Outstanding Teacher Award, Wilkes College, 1986

Institutional and Professional Service of the last five years:
Chair, RHIT Banner Web faculty Team, 2001-03
Chair, RHIT Course Management System evaluation and Selection Team, 2003-04
Leader, RHIT Implementation, Education, and Support Team, 2004-05
Freshman and New Grad Student Computer Orientation coordinator, 1999-2004

RHIT Leaves Committee, 2002-2004
Employee Relations Committee, 2001-02
Quality of Education Committee, 2000-2001
Electronic Portfolio Rater, 2001-2003
Operation Catapult Faculty, 2001-2004
Selection Committee for Computing Center Director, 2001-2002
Served on a Math Department team to design and standardize the Discrete Math sequence, 2002-03.
Led an inter-departmental team that examined the math requirements for CS majors, 2003-04.
Part of the design teamed that planned our new intro courses, did the first implementations of 220 and 230.

Professional Development Activities of the last five years:

Spent my 2005-06 sabbatical learning (by doing) about large-scale web development, .NET, databases, UI design, quality assurance.
Attended SIGCSE 2004
Attended OOPSALA 2002 and participated in the Educators' Symposium.
Attended Microsoft TechEd 2001.
Attended the ANGEL Users conferences in 2004, 2005, and 2006.
Reviewed approximately 15 textbooks for various publishers.

Faculty Curriculum Vitae - Computer Science and Software Engineering Department

Name and Academic Rank:
Mark A. Ardis, Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	1980	Computer Science	University of Maryland
Masters	1977	Computer Science	University of Maryland
B.S.	1971	Mathematics	Cornell University

Rose-Hulman Service:

Years of Service at Rose-Hulman:	6 years
Original Appointment Date and Rank:	September, 2000, Professor (untenured)
Date(s) of Promotion(s) and Resulting Rank(s):	September, 2005, Tenured Professor

Professional Experience:

2000 – Present	P**rofessor** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN
1991 – 2000	**Member of Technical Staff** Bell Labs, Naperville, IL
1987 – 1991	**Senior Computer Scientist** Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA
1983 – 1987	**Associate Professor of Information Technology** Wang Institute of Graduate Studies, Tyngsboro, MA
1980 – 1983	**Assistant Professor of Computer Science** University of Illinois, Urbana, IL
1974 – 1976	**Member of Technical Staff** Delta Research Corporation, Arlington, VA
1972 – 1974	**Consultant** Macro Systems Incorporated, Silver Spring, MD
1971 – 1972	**Associate Member of Technical Staff** Computer Sciences Corporation, Silver Spring, MD

Consulting Activities:

2000 - 2002	Innovation Fellow, Rose-Hulman Ventures, Terre Haute, IN

Patents and Disclosures:

Patent 6,580,439: (with Robert Olsen and Paul Pontrelli) Graphical user interface for providing component relational information for a uniform configuration controller for replicated component systems, 2003.

Patent 6,591,373: (with Robert Olsen and Paul Pontrelli) Uniform configuration controller for replicated component systems, 2003.

Professional Registration: None.

Principal Publications of the last five years:

Proceedings of IFIP WG8.6 Working Conference on Diffusing Software Product and Process Innovations, M. Ardis and B. Marcolin, (editors), Kluwer Academic Publishers, 2001.

"Software Engineering Baccalaureate Programs in the United States: an Overview", D. Bagert and M. Ardis, *33rd ASEE/IEEE Frontiers in Education Conference*, November 2003.

"Automating the Process of Assigning Students to Cooperative-Learning Teams", R. Cavanaugh, M. Ellis, R. Layton, and M. Ardis, *2004 American Society for Engineering Education Annual Conference & Exposition*, June 20-23, 2004.

"Test-First Teaching: Extreme Programming Meets Instructional Design in Software Engineering Courses", M. Ardis and C. Dugas, *34th ASEE/IEEE Frontiers in Education Conference*, October 20-23, 2004.

"An Incomplete History of Master of Software Engineering Programs in the United States", *15th Reunion of CMU MSE Program*, August 6, 2005.

"Diversity of Interaction in a Quality Assurance Course", M. Ardis and C. Dugas, *35th ASEE/IEEE Frontiers in Education Conference*, October 19-22, 2005.

Membership in Scientific and Professional Societies:

Association for Computing Machinery (ACM)
IEEE Computer Society
American Society for Engineering Education (ASEE)

Selected Honors and Awards:

First recipient of Nancy Martin Award for Excellence in Teaching, Wang Institute, 1985.
First recipient of Excellence in Teaching Award, CMU Master of Software Engineering Program, 1991.
PAVE Award for Technology Transfer Symposium, Lucent Technologies, 1996.
PAVE Award for Technology Transfer Symposium, Lucent Technologies, 1998.
PAVE Award for Contributions to Domain Engineering, Lucent Technologies, 1998.

Institutional and Professional Service of the last five years:

General Co-Chair and Program Chair for WWW@10 Conference, 2004
CCSE Foundations Knowledge area for Computing Curricula 2001
Team Leader for IEEE Certified Software Development Professional Exam
Reviewer for ACM, ASEE and IEEE conferences and journals
Chair of Search Committee for CSSE Department Head, Rose-Hulman Institute
Secretary of Academic Computing Committee, Rose-Hulman Institute
Chair of Advisory Committee on Computer Use, Rose-Hulman Institute
Member of Faculty Affairs Committee, Rose-Hulman Institute
Reviewer of ABET Accreditation Materials, Rose-Hulman Institute

Professional Development Activities of the last five years:

Managed software engineering projects at Rose-Hulman Ventures
Co-chair of interdisciplinary conference on the World Wide Web at Rose-Hulman (WWW@10)
Contributed software engineering education materials to online library at SWENET Workshop, 2005
Published articles (and presentations) in software engineering education

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Donald J. Bagert, Jr., Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	1986	Computer Science	Texas A&M University
M.S.	1979	Computer Science	University of Southwestern Louisiana[1]
B.S.	1977	Engineering	Tulane University

Rose-Hulman Service:

Years of Service at Rose-Hulman:	4 years
Original Appointment Date and Rank:	August 2002, Professor
Date(s) of Promotion(s) and Resulting Rank(s):	None

Professional Experience:

2002 – Present	**Professor and Director of Software Engineering** Department of Computer Science and Software Engineering Rose-Hulman Institute of Technology, Terre Haute, IN
1988 – 2002	**Assistant/Associate/full Professor** **(Associate Chair 1999-2001)** Department of Computer Science Texas Tech University, Lubbock, TX
1986 – 1988	**Assistant Professor** Department of Computer Science Northeast Louisiana University[2], Monroe LA
1980 – 1986	**Instructor** Department of Computer Science Texas A&M University, College Station, TX
1979 – 1980	**Instructor** Department of Computer Science University of Southwestern Louisiana[1], Lafayette, LA

Consulting Activities:

A Detailed Graphical Hierarchy of Electronic Information Desired for a Software Product, project for *ASEE-NAVY Summer Faculty Research Program*, 1 June to 7 August 1992, Naval Air Warfare Center-Aircraft Division, Warminster, Pennsylvania, USA.

Member, External Advisory Board, National Science Foundation Division of Undergraduate Education grant DUE #9752482 to the University of Southwestern Louisiana, 1 July 1998-30 June 2000.

Patents and Disclosures: None.

Professional Registration: Texas, since 1988
Also an IEEE Computer Society *Certified Software Development Professional* (roughly the same level as a PE, but for software professionals)

1 The University of Southwestern Louisiana is now called The University of Louisiana at Lafayette.

2 Northeast Louisiana University is now called The University of Louisiana at Monroe.

Principal Publications of the last five years:

D. J. Bagert and S. A. Mengel, "Developing and Using a Web-Based Project Process throughout the Software Engineering Curriculum", Journal of Systems and Software, Vol. 74, No. 2, pp. 113-120, 2005.

D.J. Bagert and X. Mu, "Current State of Software Engineering Master's Degree Programs in the United States". *Proceedings of the Frontiers in Education Conference*, Indianapolis, Indiana, USA, 19-22 October 2005, pp. F1G1 to F1G6.

D.J. Bagert and S. V. Chenoweth, "Future Growth of Software Engineering Baccalaureate Programs in the United States", *Proceedings of the ASEE Annual Conference*, Portland, Oregon, USA, 12-15 June 2005, CD-ROM, 8 pp.

D.J. Bagert, "Licensing and Certification of Software Professionals", Advances in Computers, Vol. 60, Academic Press, pp. 1-34, Apr. 2004. Engineering Significance: This is most comprehensive overview of licensing and certification of software professionals to date; covering the topic from an international viewpoint and including several different types of certification mechanisms.

H. Saiedian, D. J. Bagert, and N. R. Mead, "Software engineering programs: dispelling the myths and misconceptions", IEEE Software, Vol. 19, No. 4, pp. 35-41, Sept. 2002.

D.J. Bagert, "Education and training in software engineering", Encyclopedia of Software Engineering, Second Edition, John Wiley and Sons, New York, 2002, pp. 452-465.

E. Fuentetaja and D.J. Bagert, "Software Evolution from a Time-Series Perspective", *Proceedings of the International Conference on Software Maintenance*, Montreal, Canada, 3-6 October 2002, pp. 226-229.

D.J. Bagert, M. Barbacci, D. Budgen, T.C. Lethbridge, W. Suryn and H. van Vliet, "Thoughts on Software Engineering Knowledge, and How to Organize It", *Proceedings of the Conference on Software Technology and Engineering Practice*, Montreal, Canada, 6-8 October 2002, pp. 24-35.

Membership in Scientific and Professional Societies:

Association for Computing Machinery (ACM)
Institute of Electrical & Electronics Engineers (IEEE) and its Computer Society (IEEE-CS)
National Society of Professional Engineers (NSPE)
American Society for Engineering Education (ASEE)

Selected Honors and Awards:

IEEE-CS Golden Core Member, awarded May 2003
IEEE-CS Outstanding Contribution Award, 2002
Order of the Engineer, inducted 18 December 1998
IEEE Senior Member status, since 8 August 1998
Upsilon Pi Epsilon, inducted 2 December 1981

Institutional and Professional Service of the last five years:

Member, ABET Engineering Accreditation Commission, since 2006
Associate Editor in Chief for Education and Training, *IEEE Software*, since 2003
Managing Editor, *Forum for Advancing Software engineering Education* (FASE) electronic newsletter, 1997-2002 and 2003-04 (still on editorial board)
Steering Committee Chair, IEEE-CS Conference on Software Engineering Education and Training (CSEE&T), 2000-05 (now Past Chair)
Chair, IEEE Computer Society Certified Software Development Professional (CSDP) Certification Committee 1 January 2003 to 31 December 2004 (still on committee)
Member, IEEE Computer Society Educational Activities Board, since 2000
Member, IEEE Computer Society Software Engineering Program Oversight Committee (SEPOC), since 2005

Professional Development Activities of the last five years:

Attended numerous conferences and workshops in various areas of computer science and software engineering
Had published 28 journal articles, conference papers and book chapters

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Matthew R. Boutell, Assistant Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	2005	Computer Science	University of Rochester
M.S.	2002	Computer Science	University of Rochester
M.Ed.	1994	Education	University of Massachusetts
B.S.	1993	Mathematical Science	Worcester Polytechnic Institute

Rose-Hulman Service:

Years of Service at Rose-Hulman:	0.5 years (first year)
Original Appointment Date and Rank:	August 15, 2005, Assistant Professor

Professional Experience:

2005 – Present	**Assistant Professor** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN
2001 – 2005	**Research Intern** Foundation Science Center Eastman Kodak Company, Rochester, NY
2002 - 2005	**Research Assistant** Department of Computer Science University of Rochester – Rochester, NY
2000 - 2002	**Teaching Assistant** Department of Computer Science University of Rochester – Rochester, NY
1999 - 2000	**Adjunct Professor of Computer Science** Department of Computer Science Stonehill College, Easton, MA
1994 - 2000	**Mathematics and Computer Science Teacher** Department of Mathematics Norton High School, Norton, MA

Consulting Activities: P&L E-Communications, Rochester, NY – consulted on a prototype video recognition system, 2005.

Patents and Disclosures: 3 applications filed by Eastman Kodak Company to USPTO.

Professional Registration: None.

Principal Publications of the last five years:

Matthew Boutell, Jiebo Luo, and Robert T. Gray. Sunset scene classification using simulated image recomposition. *IEEE International Conference on Multimedia and Expo*, Baltimore, MD, July 2003.

Matthew Boutell and Jiebo Luo. Bayesian fusion of camera metadata cues in semantic scene classification. *IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, June 2004.

Matthew Boutell, Jiebo Luo, and Christopher Brown. Learning spatial configuration models using modified Dirichlet priors. *Workshop on Statistical Relational Learning (in conjunction with ICML2004)*, Banff, Alberta, July 2004.

Matthew Boutell, Xipeng Shen, Jiebo Luo, and Christopher Brown. Learning multi-label semantic scene classification. *Pattern Recognition*, 37(9), pp. 1757-1771, September 2004.

Jiebo Luo and Matthew Boutell. Automatic image orientation detection via confidence-based integration of low-level and semantic cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5), pp. 715-726, May 2005.

Jiebo Luo, Matthew Boutell, Robert T. Gray, and Christopher Brown. Image transform bootstrapping and its applications to semantic scene classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(3), June 2005.

Matthew Boutell, Jiebo Luo, and Christopher Brown. A generalized temporal context model for classifying image collections. *ACM Multimedia Systems*, 11(1), pp. 82-92, November 2005.

Membership in Scientific and Professional Societies:

IEEE

IEEE Computer Society

Selected Honors and Awards:

B.S. degree from WPI with High Distinction

Supported by a federal GAANN fellowship for first three years of PhD studies.

Award for most papers published by a graduate student, URCS, 2004.

Best Student Paper award, *2004 IEEE Western New York Image Processing Workshop*

Included three times in *Who's Who Among American High School Teachers*

Inducted into *Pi Mu Epsilon* and *Tau Beta Pi* honor societies.

Won *Worcester Polytechnic Institute Class of 1879 Prize for Outstanding Project in the Humanities* for original composition for brass quintet.

Institutional and Professional Service of the last five years:

Organized bi-weekly Vision Interest Meetings at URCS, 2004-5.

Graduate Representative to the URCS Faculty, 2003-4.

Graduate Admissions Committee, URCS, 2002-3.

Professional Development Activities of the last five years:

Program committee, Workshop on Semantic Learning Applications in Multimedia, *IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, 2006.*

Reviewed papers for *Pattern Recognition, IEEE Signal Processing Magazine, SPIE Journal of Electronic Imaging*.

Attended 5 major conferences and numerous workshops, presenting at 12 of them.

Faculty Curriculum Vitae –
Department of Computer Science and Software Engineering

Name and Academic Rank:
Steve Chenoweth, Associate Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	1990	Computer Science & Engineering	Wright State University
M.S.	1986	Computer Science	Wright State University
MBA	1980	Management	Wright State University
M.S.	1971	Education	Butler University
B.A.	1966	Mathematics	Butler University

Rose-Hulman Service:

Years In Service at Rose-Hulman:	2 ½
Date of Original Appointment to Rose-Hulman:	2003
Dates of any Advancement in Rank at Rose-Hulman:	None

Professional Experience:

1995-2002	Member of Technical Staff, Bell Laboratories, Lucent Technologies and AT&T, Columbus, OH,
1974-1995	Consulting Analyst, Senior Systems Analyst, and Manager. NCR Corporation, Dayton, OH.

Consulting Activities:

2002-2003	Millennium Services, Lincroft, NJ – Software architecture and process consulting.
2005	Consulted in Indianapolis, for one of the client companies of Rose-Hulman Ventures.
2005	Worked with Rose-Hulman Ventures, to develop related CSSE classroom activities related to their work.

Patents:
1. Multi-function product label - Assigned to NCR, 1999 (US patent 6,019,394).
2. Parallel processing using competing algorithms - Assigned to NCR, 1997 (US patent 5,689,631).

State(s) in which registered:
None

Principal Publications of the Last Five Years:
"Teamwork and Robotics – A Leadership Experience for Undergraduate Engineering Students," by Steve Chenoweth and David Mutchler. To be presented at ASEE IL-IN, 2006.

"Methods of Teaching Leadership to Undergraduate Students in Computer Science and Software Engineering," by Steve Chenoweth. Paper and presentation at ASEE IL-IN, March, 2005.

"Future Growth of Software Engineering Baccalaureate Programs in the United States," by Donald J. Bagert and Stephen V. Chenoweth. Paper was accepted and presented at the national ASEE conference by Dr. Bagert, June, 2005.

"Project Management: Electrical Engineering vs. Software Engineering," by Steve Chenoweth and Mark A. Yoder. ASEE Peoria Regional Conference, March, 2004.

"Giving Students Experience in Dealing with Clients In a Course on Requirements," by Donald J. Bagert and Stephen V. Chenoweth. ASEE GSW Regional Conference, March, 2004.

Membership in Scientific and Professional Societies:

IEEE, IEEE Computer Society, ACM SIGCSE, ASEE

Selected Honors and Awards:

None

Institutional and Professional Service in the last Five Years:

Member of committee to create a Masters of Software Engineering program at RHIT.
One of four members of the Software Engineering curriculum team, 2003-4 and 2004-5.
Member of Graduate Studies Committee, 2004-5.
Initiated and directed a monthly department e-newsletter to prospective Rose students.
Taught a session of RHIT's Catapult program for high school students, 2004 and 2005.
Academic adviser for up to 20 CS majors during the past two years.

Professional Development Activities in the Last Five Years:

Served as a referee for papers at ASEE Indiana / Illinois regional conference, 2005.

Co-provider of workshops on teaching software architecture at CSEE&T (Conference on Software Engineering Education & Training, 2004 and 2005.

Initiated a program to evaluate the effectiveness of the courses of two courses (CSSE 120 and 374), taught in 2004 and 2005.

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Archana Chidanandan, Assistant Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	2004	Computer Engineering	University of Louisiana at Lafayette
Masters	1999	Computer Engineering	University of Louisiana at Lafayette
B.E.	1997	Electronics and Communications Engg.	Anna University, India

Rose-Hulman Service:

Years of Service at Rose-Hulman:	3 years
Original Appointment Date and Rank:	April, 2003, Assistant Professor

Professional Experience:

2003 – Present	**Assistant Professor** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN
1997 – 1998	**Assistant Systems Engineer** Tata Consultancy Services, Madras, India

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: None

Principal Publications of the last five years:

- A. Chidanandan, J. Moder and M. Bayoumi, "Implementation of NEDA-based DCT architecture using even-odd decomposition of the 8 x 8 DCT matrix," submitted to the Midwest Symposium on Circuits and Systems (MWSCAS 2006).
- A. Chidanandan and M. Bayoumi, "Area-efficient NEDA Architecture for the 1-D DCT/IDCT," 2006 IEEE International conference on Acoustics, Speech, and Signal Processing (ICASSP06), Toulouse, France
- A. Chidanandan and M. Bayoumi,**"** Novel Systolic Array Architecture for the Decorrelator using Conjugate Gradient for Least Squares Algorithm," IEEE Symposium on Circuits and Systems (ISCAS 2005), Kobe, Japan
- Ahmed Shams, Archana Chidanandan, Wendi Pan and Magdy Bayoumi,``NEDA: A new low power high performance DCT architecture," accepted for publication to the IEEE Transactions in Signal Processing.
- Chidanandan and M. Bayoumi, "Enhanced Parallel Interference Cancellation using Decorrelator for the Base-Station Receiver," IEEE Symposium in Circuits and Systems (ISCAS 04), Vancouver, Canada
- P. Zhao, P. Golconda, A. Chidanandan, and M. Bayoumi,"A Double Edge Implicit Pulsed Level Convert Flip-Flop for Level Conversion In CVS Systems", ISVLSI 2004 IEEE CS conference on VLSI 2004, Lafayette, LA
- Shams, W. Pan, A. Chidanandan, M. Bayoumi "A Low Power High Performance Distributed DCT Architecture," ISVLSI 2002 IEEE CS conference on VLSI 2002, Philadelphia, PA

Membership in Scientific and Professional Societies:
Institute of Electrical and Electronic Engineers (IEEE)
Association of Computer Machinery(ACM)

Institutional and Professional Service of the last five years:

- Member of the Graduate Studies Committee, August 2005 – current

- Member of the Quality of Education Committee, August 2004 – May 2005
- Member of the Visual Arts and Performing Committee, August 2003- May 2004
- Member of the group that worked on the proposal for the Masters in Software Engineering program
- Instructor for Operation Catapult 2005. Worked with students on JAVA programming projects.
- Serve as Academic Advisor to Computer Science and Software Engineering students.
- Served as Freshman Advisor, August 2004-May 2005
- Faculty advisor to a new club called "gEECS" – girls in Electrical Engineering, Computer Science and Software Engineering. The group is applying for club status this year.
- Guided a group of four Computer Science and Computer Engineering female students as they participated in the "Games 4 Girls" programming competition organized by the University of Illinois at Urbana-Champagne. The students were awarded the third prize at the competition.
- Received a CREU grant from CRA-W along with group of two computer engineering students to work an image compression project, 2004-2005.
- Received a CRA-W grant to host a distinguished speaker.
- Hosted a guest speaker from AOL.
- Served as host to a guest speaker at the WWW@10 conference.
- Served on the faculty search committee in the CSSE department, 2004-2005.
- "Introduction to today's Computer Networks", presentation made to the Systems Analysis and Design class at St. Mary of the Woods College, November 2005.

Professional Development Activities of the last five years:

- Program committee member, Indiana Women in Computing Conference, InWiC 2006
- Technical program committee member, IEEE Workshop on Signal Processing Systems,2005(SiPS 05)
- Reviewer for IEEE Symposium on Circuits and Systems 2005 (ISCAS 2005), 2004, 2003, American Association for Engineering Education Conference 2005 (ASEE 2005), 10th Asia-Pacific Computer Systems Architecture Conference, 2005, IEEE Conference on Acoustics, Signals and Speech Processing, 2004,2003.
- Evaluator of Student Research Competition submissions at SIGCSE, 2005
- Panel member at a session at the Central Indiana Celebration of Women in Computing Conference, CICWIC 04, Feb. 2004.
- Attended three workshops titled "Active and Co-operative Learning Techniques", "From Nand to Tetris" and "Facilitating student-written Operating Systems"SIGCSE 2005, St. Louis, MO
- Attended workshop titled "Managing the Academic Career for Faculty Women at Undergraduate Computer Science and Engineering Institutions" – sponsored by Computer Research Association (CRA-W), St. Louis, MO
- Attended workshop titled "Assessment of Student Writing in the Classroom and Across Programs: an Internal and Outcomes Improvement Approach", Best Assessment Processes VII Symposium, Terre Haute, IN
- Attended seminars titled "Using Linux in Windows Environments", and "Linux as an alternative desktop", Illianatech SummIT 2005, Terre Haute, IN
- Attended seminars sponsored by AVID Technolgy, Inc and Mentor Graphics, Indianapolis, IN, May 2005.
- Attended seminars on use of peer-evaluation to assess student work, effective and efficient teaching, use of Dyknow technology in the classroom, place for HSS in the computing and engineering curriculum, best classroom teaching practices – organized by the Quality of Education committee at Rose-Hulman

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Curtis Clifton, Assistant Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	2005	Computer Science	Iowa State University
Masters	2001	Computer Science	Iowa State University
B.S.E.E.	1992	Electrical Engineering	Iowa State University

Rose-Hulman Service:

Years of Service at Rose-Hulman:	first
Original Appointment Date and Rank:	August, 2005, Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s):	not applicable

Professional Experience:

2005 – Present	**Asst. Professor, Computer Science and Software Engineering Department** Rose-Hulman Institute of Technology, Terre Haute, IN
1998 – 2005	**Research and Teaching Assistant** Iowa State University, Ames, IA
1995 – 1998	**Software Team Leader and Sr. Electrical Engineer** L&S Electric, Inc., Wausau, WI
1992 – 1995	**Manufacturing Manager** Procter & Gamble, Inc., Iowa City, IA

Consulting Activities:

1998 – 1999	**Independent Consultant** Hydroelectric plant automation

Patents and Disclosures: None.

Professional Registration: None.

Principal Publications of the last five years:

Journal Papers

1. Curtis Clifton and Gary T. Leavens. MiniMAO1: An Imperative Core Language for Studying Aspect-Oriented Reasoning. Science of Computer Programming, 2006. To appear.
2. Curtis Clifton, Todd Millstein, Gary T. Leavens, and Craig Chambers. MultiJava: Design rationale, compiler implementation, and applications. ACM Trans. on Prog. Lang. and Sys., 2006. To appear. Pre-print available as ISU Tech. Report #04-01b.
3. Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the design of JML accommodates both runtime assertion checking and formal verification. Science of Computer Programming, 55(1-3):185–208, March 2005.

Papers at Refereed Conferences

1. Gary T. Leavens and Curtis Clifton. Lessons from the JML Project. In Verified Software: Theories, Tools, Experiments. Zurich, Switzerland. IFIP Working Group 2.3, October 2005.
2. Gary T. Leavens, Yoonsik Cheon, Curtis Clifton, Clyde Ruby, and David R. Cok. How the design of JML accommodates both runtime assertion checking and formal verification. In Frank S. de Boer, Marcello M. Bonsangue, Susanne Graf, and Willem-Paul de Roever, editors, Formal Methods for Components and Objects: First International Symposium, FMCO 2002, Lieden, The Netherlands, November 2002, Revised Lectures, volume 2852 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2003.

3. Curtis Clifton, Gary T. Leavens, Craig Chambers, and Todd Millstein. MultiJava: Modular open classes and symmetric multiple dispatch for Java. In OOPSLA 2000 Conference on Object-Oriented Programming, Systems, Languages, and Applications, volume 35(10) of ACM SIGPLAN Notices, pages 130–145, New York, October 2000. ACM.

Papers Presented at Refereed Workshops

1. Curtis Clifton and Gary T. Leavens. MiniMAO: Investigating the semantics of proceed. In Curtis Clifton, Gary T. Leavens, and Ralf Lämmel, editors, FOAL 2005 Proceedings: Foundations of Aspect-Oriented Languages Workshop at AOSD 2005, number 05-05 in Computer Science Technical Reports, pages 51–62. Department of Computer Science, Iowa State University, March 2005.
2. Curtis Clifton and Gary T. Leavens. Obliviousness, modular reasoning, and the behavioral subtyping analogy. In SPLAT 2003: Software engineering Properties of Languages for Aspect Technologies at AOSD 2003, March 2003. Available as Computer Science Technical Report TR03-01a from ftp//:ftp.cs.iastate.edu/pub/techreports/TR03-01/TR.pdf.
3. Curtis Clifton and Gary T. Leavens. Observers and assistants: A proposal for modular aspect-oriented reasoning. In Gary T. Leavens and Ron Cytron, editors, FOAL 2002 Proceedings: Foundations of Aspect-Oriented Languages Workshop at AOSD 2002, number 02-06 in Computer Science Technical Reports, pages 33–44. Department of Computer Science, Iowa State University, April 2002.

Membership in Scientific and Professional Societies:

Association for Computing Machinery (ACM) – special interest groups in Computer Science Education (SIGCSE), Programming Languages (SIGPLAN), and Software Engineering (SIGSOFT)
American Society for Engineering Education (ASEE)
Institute of Electric and Electronics Engineers (IEEE) – IEEE Computer Society

Selected Honors and Awards:

Graduate Scholarship, $900 per semester, Fall 2003, Spring 2004
Preparing Future Faculty Associate, 2002
Iowa State University Research Excellence Award, 2001

Institutional and Professional Service of the last five years:

Foundations of Aspect-Oriented Languages (FOAL) Workshop, Co-chair, 2003–2006
Iowa State Computer Science Graduate Student Organization, President and founder, 1999–2002
Dept. Chairperson Search Committee, Student representative, 2001
Reviewer for: Object-Oriented Programs, Systems, Languages, and Applications (OOPSLA), 2004; Trans. on Prog. Lang. and Systems (TOPLAS), 2003; European Conf. on Object-Oriented Prog. (ECOOP), 2003; FOAL, 2004, 2006; Aspect-Oriented Software Development (AOSD), 2003

Professional Development Activities of the last five years:

Attended conferences and workshops: OOPSLA, 2000–2005; OOPSLA Educator's Symposium, 2000–2002, 2004; Extravagaria workshop, 2003; AOSD, 2002–2004, 2006; FOAL, 2002–2004, 2006

Name and Academic Rank:
Lisa C. Kaczmarczyk, Assistant Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	2005	Interdisciplinary	University of Texas at Austin
Masters	1992	Computer Science	University of Oregon
Masters	1990	Information Systems	Northeastern University
B.S.	1983	Drama, Spanish	Tufts University

Rose-Hulman Service:

Years of Service at Rose-Hulman:	1 year
Original Appointment Date and Rank:	August, 2005, Assistant Professor

Professional Experience:

1999 – 2005	**Assistant Instructor** Department of Computer Sciences University of Texas at Austin, Austin, TX
Summer 2001	**Visiting Lecturer** University of Oregon, Eugene, OR
1992 – 1999	**Instructor** Chemeketa Community College, Salem, OR
1999 – 2000	**Software developer** Pointserve Inc, Austin, TX

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: None.

Principal Publications of the last five years:

Kaczmarczyk, L. C., Last, M. Z, Miikkulainen, R. (in press). "The Effect of Delivery Method on Conceptual and Strategy Development". To appear in *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, Vancouver, British Columbia.

Kaczmarczyk, L.C., Miikkulainen, R. (2004). "The Acquisition of Intellectual Expertise: A Computational Model". *Proceedings of the 26th Annual Conference of the Cognitive Science Society*, Chicago, Illinois.

Kaczmarczyk, L. C. (2003). "A Technical Writing Class for Computer Science Majors: Measuring Student Perceptions of Learning." *Proceedings of the 34th Annual Technical Symposium on Computer Science Education (SIGCSE 2003)*, Reno, Nevada.

Kaczmarczyk, L. C. (2001a, June). "Accreditation and student assessment: why we all need to pay attention." *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2001)*, Canterbury, UK.

Membership in Scientific and Professional Societies:

Cognitive Science Society
Association of Computing Machinery (Special Interest Group on Computer Science Education)
IEEE Computer Society

Selected Honors and Awards:

Invited Keynote Speaker, Indiana Women in Computing Conference, 2006
Awarded scholarship to Grace Hopper Conference on Women in Computer Science, 2004
Awarded scholarship to Grace Hopper Conference on Women in Computer Science, 2002

Institutional and Professional Service of the last five years:

Reviewer for the National Science Foundation – Multiple occasions

Invited Talk: "The Acquisition of Intellectual Expertise: A Computational Theory." (2003, January) School of Computing, University of Kent, Canterbury, UK.

Invited Talk: "Neural Networks and Modeling of Student Learning." (2001, June) School of Computing, University of Leeds, UK.

Invited Article: Kaczmarczyk, L. C. (2001b, December). "Curriculum descant: Is AI abstract and impractical? Isn't the answer obvious?" Intelligence, 12, 19-20.

Kaczmarczyk (Moderator), Kruse, Lopez, Kumar (2004) "Incorporating Writing into the CS Curriculum." *Proceedings of the 35th Annual SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2004)*, Norfolk, Virginia.

Professional Development Activities of the last five years:

Attended Frontiers in Education Conference, Indianapolis, IN., 2005.
Coordinator (co) of Student Volunteers, SIGCSE Conference, 2006, Houston, TX.
Attended Computing Research Association Academic Careers Workshop, Washington DC, 2006.
Attended first International Computing Education Research Workshop (ICER), 2005, Seattle, WA.

Faculty Curriculum Vitae - Computer Science and Software Engineering Department

Name and Academic Rank:

J.P. Mellor, Associate Professor of Computer Science and Software Engineering.

Degrees:

Ph.D.	2000	Electrical Engineering and Computer Science	Massachusetts Institute of Technology.
S.M.	1995	Electrical Engineering and Computer Science	Massachusetts Institute of Technology.
B.S.	1990	Nuclear Technology	University of the State of New York.
B.S.	1987	Electrical Engineering	United States Naval Academy.

Rose-Hulman Service:

Years of Service at Rose-Hulman:	7 years.
Original Appointment Date and Rank:	November, 1999, Assistant Professor.
Date(s) of Promotion(s) and Resulting Rank(s):	September, 2005, Associate Professor.

Professional Experience:

2002–Present	**Director, Imaging Systems Laboratory** Rose-Hulman Institute of Technology, Terre Haute, IN.
1999–Present	**Asst. Professor/Assoc. Professor** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN.
2001–2005	**Innovation Fellow & Project Manager** Rose-Hulman Ventures, Terre Haute, IN.
1997–1999	**System Administrator** Zoesis Studios, Newton, MA.
1996–1999	**Custom Software Developer** **Advanced Unix System Administration Instructor** Great Eastern Technology, Woburn, MA.
1992–1993	**Material Assistant** Submarine Squadron Three, San Diego, CA.
1990–1992	**Main Propulsion Assistant** U.S.S. Drum, SSN-677, San Diego, CA.
1989–1990	**Shift Engineer/Assistant Shift Engineer** Naval Nuclear Power Training Unit, Ballston Spa, NY.

Consulting Activities:

2000–2001	**Expert Witness** Kirkland & Ellis Law Firm, Chicago, IL.

Patents and Disclosures:

September 2002 System and Method for Transforming Graphical Images.

Professional Registration:

1992 Certified as Chief Engineer of Naval Nuclear Propulsion Plants.

1988 Certified for supervision, operation, and maintenance of Naval Nuclear Propulsion Plants.

Principal Publications of the last five years:

"Preliminary Findings on the Clinical Utility of an Unobtrusive Home Health Monitoring System," P.A. Woodbridge, J.P. Mellor and M. Weiner, *Human-Computer Interaction International*, July 2005.

"An Improved Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, *Workshop on Applications of Computer Vision*, January 2005.

"An Improved Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, Sandor Pethes, Joshua Starr, Kevin Gorsky, and Charles Lehman, Technical Report RHV-TR-2, Rose-Hulman Institute of Technology, Rose-Hulman Ventures, September 2003.

"An Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, *British Machine Vision Conference*, September 2003.

"New Approach to Creating a Successful Technology-Based Incubator," Brij Khorana, John Davidson, J.P. Mellor, National Business Incubator Association, May 2003.

"Geometry and Texture from Thousands of Images," J.P. Mellor, *International Journal of Computer Vision*, 51(1):5–35, January 2003.

"Engineering Education For Inclusive Design," Lawrence Goldberg, Eric Jolly, J.P. Mellor, Babette Moeller, Madeleine Rothberg, Richard Stamper, Michael Wollowski, *Frontiers in Education*, November 2002.

"An Image-Based Three-Dimensional Digitizer for Pre-Decorating Thermoformed Parts," J.P. Mellor, Mike Oder, Joshua Starr, and Joel Meador, Technical Report RHV-TR-1, Rose-Hulman Institute of Technology, Rose-Hulman Ventures, April 2002.

Membership in Scientific and Professional Societies:

Institute of Electrical and Electronics Engineers (IEEE)
Association for Computing Machinery (ACM)
Sigma Xi, The Scientific Research Society
Tau Beta Pi

Selected Honors and Awards: None

Institutional and Professional Service of the last five years:

Imaging Systems Certificate Advisor, 2002–present.
Program committee member for the WWW@10: The Dream and the Reality conference, 2004.
Faculty advisor for the Rose-Hulman Computing Society.
Faculty teller, 2002-2004.

Professional Development Activities of the last five years:

Created several new courses including Computer Vision and Advanced Operating Systems.
Initiated major revisions to the Computer Architecture sequence.
Published articles (and presentations) in the area of computer science.
Participated in several research grants.
Assisted several clients of Rose-Hulman Ventures.

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Laurence D. Merkle, Assistant Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	1996	Computer Engineering	Air Force Institute of Technology
M.S.C.E.	1992	Computer Engineering	Air Force Institute of Technology
B.S.	1987	Computers & Systems Engineering	Rensselaer Polytechnica Institute

Rose-Hulman Service:

Years of Service at Rose-Hulman:	4 years
Original Appointment Date and Rank:	September, 2002, Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s):	N/A

Professional Experience:

2002 – Present	**Assistant Professor** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN
1999 – 2002	**Assistant/Associate Professor** Computer Science Department United States Air Force Academy, Colorado Springs, CO
1997 – 1998	**Adjunct Professor** Chapman University and College of Santa Fe, Albuquerque, NM
1996 – 1999	**Chief, Center for Plasma Theory and Computation** Air Force Research Laboratory, Albuquerque, NM
1988 – 1991	**Artificial Intelligence Project Officer** Air Force Materiel Command, Dayton, OH

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: None

Principal Publications of the last five years:

"Underwater Hacker Missile Wars: A Cryptography and Engineering Contest," Joshua Holden, Richard Layton, Laurence Merkle, and Tina Hudson, Cryptologia, Vol. 30, (pp. 69-77), 2006.

"Multi-Robot Cooperation Using the Ant Algorithm with Variable Pheromone Placement," Eric Borzello and Laurence Merkle, proceedings of the IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, 2005.

"Evolutionary Computation in Polymorphous Computing Architectures," Laurence D. Merkle, Matthew G. Ellis, and Michael C. McClurg, presented at the DARPA Polymorphous Computing Architectures Principal Investigators' Meeting, Boulder, Colorado, August 2005.

"Underwater Model Rockets: An Innovative Design Problem and Competition for Undergraduate Students in Engineering, Math and Science," Richard Layton, Josh Holden, Tina Hudson, and Laurence Merkle, proceedings of the ASEE Annual Conference, Session #3566, Portland, Oregon, June 12-15, 2005. Best paper in Mechanical Engineering Division.

"Hybrid Architectures for Evolutionary Computing Methods: Automated Transfer of Evolutionary Computation Successes to the Evolvable Hardware Domain," Laurence D. Merkle and Daniel J. Burns, presented in the Military and Security Applications of Evolutionary Computation Workshop of the Genetic and Evolutionary Computation Conference, Seattle, Washington, 2004.

"Automated Load Balancing of a Missile Defense Simulation Using Domain Knowledge," Martin C. Carlisle and Laurence D. Merkle, Journal of Defense Modeling and Simulation, Vol. 1, No. 1, (pp. 59-68), April 2004.

"Design Optimization for a Novel Class of High Power Microwave Sources," Laurence D. Merkle and John W. Luginsland, invited paper for the IEEE Congress on Evolutionary Computation special session on Evolutionary Design Optimization, Canberra, Australia, 2003.

"Measuring the Effectiveness of Robots in Teaching Computer Science," Barry Fagin and Laurence Merkle, proceedings of the Technical Symposium on Computer Science Education, Reno, Nevada, 2003.

"Towards Effective Evolutionary Algorithms for Polypeptide Structure Prediction," Gary B. Lamont and Laurence D. Merkle, in "Evolutionary Computation in Bioinformatics, " ed. Gary Fogel and David W. Corne, 2002.

"Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education," Barry S. Fagin and Laurence Merkle, ACM Journal of Educational Resources in Computing, Vol. 2, No. 4, (pp. 1-18), December, 2002.

"Evolution of an Introductory Computer Science Course: The Long Haul," A. T. Chamillard and Laurence D. Merkle, proceedings of the Rocky Mountain Conference of the Consortium for Computing in Small Colleges, 2002.

Membership in Scientific and Professional Societies:

American Association for Artificial Intelligence
American Society of Engineering Education
Association of Computing Machinery
Institute of Electrical and Electronics Engineers

Selected Honors and Awards:

Best Paper, Mechanical Engineering Division, 2005 American Society for Engineering Education Annual Conference & Exposition
USAFA Department of Computer Science Research Excellence Award, 2001-2002
Upsilon Pi Epsilon, Eta Kappa Nu, Tau Beta Pi

Institutional and Professional Service of the last five years:

Committees: Faculty Affairs, Rules and Discipline, Quality of Education,
M.S. Thesis Committee Member (Dillman, Morgan, Schmitt, Yargaladda)
Faculty Advisor (Upsilon Pi Epsilon, Tau Beta Pi, Pi Kappa Alpha, Rose Chorus)
Program Committee (Genetic and Evolutionary Computation Conference (GECCO); Technical Symposium on Computer Science Education; ACM Symposium on Applied Computing; IEEE International Conference on Systems, Man and Cybernetics; IEEE Information Assurance Workshop)
Chair/co-chair, Undergraduate Workshop, GECCO
Co-chair, Military and Security Applications of Evolutionary Computation, GECCO
Web-based SIGCSE Conference Registration System

Professional Development Activities of the last five years:

Evolutionary Computation in Polymorphous Computing Architectures
Visiting Professor, Information Institute, Air Force Research Laboratory.
Information Warfare Applications Course
Program Manager for High Performance Computing, Air Force Office of Scientific Research.

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
David Mutchler, Professor of Computer Science and Software Engineering

Degrees:

Ph.D.	1986	Computer Science	Duke University
M.S.	1980	Mathematics	University of Virginia
B.A.	1976	Mathematics	University of Virginia

Rose-Hulman Service:

Years of Service at Rose-Hulman:	12 years
Original Appointment Date and Rank:	September, 1994, Associate Professor
Date(s) of Promotion(s) and Resulting Rank(s):	September, 2000, Full Professor

Professional Experience:

1994 – Present	**Assoc. Professor/Professor** Department of Computer Science and Software Engineering Rose-Hulman Institute of Technology, Terre Haute, IN
1987 – 1994	**Assistant Professor** University of Tennessee, Knoxville, TN
1986 – 1987	**Research Computer Scientist** Naval Research Laboratory, Washington, DC
1981 – 1986	**Research Assistant** Duke University, Durham, NC
1980 – 1981	**Instructor** Davidson College, Davidson, NC
1976 – 1980	**Teaching Assistant** University of Virginia, Charlottesville, VA

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: None.

Principal Publications of the last five years:

Joint with colleagues from Rose-Hulman, presented at the First Workshop on the Impact of Pen-based Technology on Education, Purdue University, April 6-7, 2006:

- "Tablet or Laptop? Does the pen matter, and if so, in what way?"
- "An interdisciplinary project using handheld, pen-based devices for project management, technical writing for mobile devices, and software development."
- "Pen-based tablet PCs with DyKnow in introductory 'studio' physics classes."
- "Implementing Tablet Computers in the technical communication classroom: measuring the impact of mobility on communication skills."

Co-PI of $100,000 funded grant from Microsoft, "Assessing and Evaluating the Symbiosis of Tablet PCs and Collaboration-Facilitating Software in the Classroom," February 2006.

Joint with colleagues from Rose-Hulman: "The world is our classroom: Tablet PCs", Hewlett-Packard University Mobile Technology Solutions Conference 2004, San Jose, California, November 4-5, 2004.

Co-PI of $220,000 and $135,500 funded grants from Hewlett-Packard, "Using Mobile Technology to Amplify Interactions and Enhance Learning in the Classroom and Laboratory," February 2003 and February 2004.

"The role of Botball teachers: a teacher-as-facilitator model," and "How and why to run a Botball 'Midway' Tournament," National Botball Robotics Education Conference, Norman, OK, June 29 - July 2, 2002.

Lynn Kiaer, David Mutchler and Jeffrey Froyd, "Laptop computers in an integrated first-year curriculum," *Communications of the ACM* 41(1), pp. 45-49 (January, 1998).

Membership in Scientific and Professional Societies:

Association of Computing Machinery (ACM). American Association for Artificial Intelligence (AAAI). Fulbright Academy . Special Interest Group for Computer Science Education (SIGCSE). Special Interest Group for Artificial Intelligence (SIGART)

Selected Honors and Awards:

Fulbright Award for Lecturing/Research, University of Mauritius, Mauritius, 2000-2001.

Outstanding Faculty Member of the Week, as selected by President Samuel F. Hulbert, Rose-Hulman Institute of Technology: Many times.

1992-93 ACM Teacher of the Year, as selected by the Univ. of Tennessee chapter of the ACM.

1989-90 ACM Teacher of the Year, as selected by the Univ. of Tennessee chapter of the ACM.

Institutional and Professional Service of the last five years:

Coordinator of the Midwest Regional Botball program, 1997 to date.

Leader of various area First Lego League teams, 1998 to date.

Co-coordinator of the Tablet PC and DyKnow program at Rose-Hulman, 2002 to date.

Member of planning and program committees for "The World Wide Web at 10: The Dream and the Reality," Rose-Hulman. October 2004.

Faculty advisor for Unity, a Rose-Hulman club whose mission is "to provide a place where individuals can be themselves without fear of prejudice or hate, and to promote an environment on campus and throughout our community that is welcoming to all people," 2001 to date.

Faculty advisor for the Board Games club, 1995 to date.

Suggested a "Safe Zone" project that the Diversity Council sponsored, implemented during 2004-2005.

Frequent leader of activities for Explore Engineering outreach program.

Professional Development Activities of the last five years:

- Conferences attended include:

National Botball Robotics Education Conference, Jacksonville, Florida, July 14-17, 2005.

The First Annual Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-2005), Marina del Rey, California, June 1-3, 2005

Hewlett-Packard University Mobile Technology Solutions Conference 2004, San Jose, CA, Nov. 4-5, 2004.

"The World Wide Web at 10: The Dream and the Reality," Rose-Hulman Institute of Technology, Oct. 2004.

Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, California, July 25-29, 2004.

2004 American Association for Artificial Intelligence (AAAI) Spring Symposium on Accessible Hands-on Artificial Intelligence and Robotics Education, at Stanford University, March 24-26, 2004.

Hewlett-Packard University Mobile Technology Solutions Conference 2003, Palo Alto, October 16-17, 2003.

Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August 9-15, 2003.

Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada, July 28 – Aug. 1, 2002.

3rd International Conference on Computers and Games, Edmonton, Alberta, Canada, July 25 - 27, 2002.

National Botball Robotics Education Conference, Norman, Oklahoma, June 29 - July 2, 2002.

Robotics Educators Workshop, Norman, Oklahoma, January 10-13, 2002.

- Talks not associated with publications listed above include:

"Using Tablet PCs creatively," Best Assessment Processes VI Symposium, Rose-Hulman, February, 2004.

"Teaching (Technically) by Storytelling" at Rose-Hulman's Opening Symposium, August 2002.

- Sabbatical:

Fulbright Award for Lecturing/Research, University of Mauritius, Mauritius, 2000-2001.

Faculty Curriculum Vitae – Computer Science and Software Engineering Department

Name and Academic Rank:
Michael Wollowski, Associate Professor of Computer Science and Software Engineering

Degrees:

Ph..D.	1998	Computer Science	Indiana University, USA
M.S.	1988	Computer Science	Indiana University, USA
Vordiplom,	1986	Informatik	Universitat Hamburg, Germany

Rose-Hulman Service:

Years of Service at Rose-Hulman:	Seven years
Original Appointment Date and Rank:	August 1999, Assistant Professor
Date(s) of promotion(s) and Resulting Rank(s)	August 2005, Associate Professor

Professional Experience:

1999 – Present	**Asst./Assoc. Professor** Computer Science and Software Engineering Department Rose-Hulman Institute of Technology, Terre Haute, IN
1995 – 1999	**Visiting Assistant Professor** Computer Science Department, Siena College, Loudonville, NY.
1990 – 1995	**Associate Instructor** Computer Science Department, Indiana University, Bloomington, IN.

Consulting Activities: None

Patents and Disclosures:
Michael Wollowski, Robert Signorelli, Chris Barrell: *A Method and System for User Initiated Repeat Purchases via the Internet* on the IP.com Prior Art Database.

Professional Registrations: None

Principal Publications of the last five years:

- Michael Wollowski. *A Theorem Prover for a Diagrammatic Blocks World.* Proceedings of The 2006 Midwest Artificial Intelligence and Cognitive Science Conference, Valparaiso, IN, April 1-2, 2006.
- Michael Wollowski. *Search and Inference with Diagrams*. Proceedings of The Ninth IASTED International Conference on INTERNET & MULTIMEDIA SYSTEMS & APPLICATIONS, Honolulu, HI, Aug 15 – 17, 2005.
- Michael Wollowski. *Living in a Transparent Future: Search in a Wired World.* Online Proceedings of the WWW@10 Conference, Terre Haute, IN, Sep 30 – Oct 2, 2004.
- Michael Wollowski, Peter Nei, Chris Barrell. *A Diagrammatic Inference System for the Web.* Proceedings of The Thirteenth International World Wide Web Conference, Alternate Track Papers & Posters, pp. 374-375. NY, NY, 2004.
- Michael Wollowski, *An XML-Based Syllabus Editor and Search Engine*. Proceedings of the Stop Surfing - Start Teaching 2003 National Conference, Las Vegas, 2003, pp 107 -

111.

- Michael Wollowski, *XML Based Course Websites*. Proceedings of the E-Learn 2002 Conference, Montreal, 2002, pp 1043 - 1048.
- Michael Wollowski, *An Undergraduate Research Course Aimed at Furthering the Web*, Proceedings of the 2001 Frontiers in Education Conference, Reno, NV, from Oct 10 - 13.

Membership in Scientific and Professional Societies:

- Association of Computing Machinery
- Web Intelligence Consortium

Selected Honors and Awards: None

Institutional and Professional Service in the last Five Years:

- Reviewer for ACM's Computing reviews
- Board member of *Engenius Solutions*.
- Faculty advisor to the Web-based Soda Machine project
- Served on several committees: Quality of Education, International Programs, and Rules and Discipline (chair), Advisory Committee of Computer Use
- Actively participated in the hiring of all of our new colleagues.
- Chaired sessions at several conferences
- Developed several new courses: CSSE 120, CSSE 220, CSSE 481, CSSE 490: Swarm Intelligence. Some of those courses were developed together with Andy Kinley and David Mutchler. Revised CSSE 230.
- Pioneered the use of the web in our courses.
- Academic advisor to about 20 students
- Gave several work-shops to students and faculty about creating web pages
- Chaired the Technology Enhanced Instruction Team, recommended directions that Rose-Hulman might employ and support in the areas of web-based instruction, technology in the classroom, and in general any other technology that might enhance our ability to provide the best learning experience possible.

Professional Development Activities in the Last Five Years:

- Attended the conferences listed under Publications
- Attended many of the Teaching Enhancement seminars given over the years here at RHIT
- Attended the *2002 Search Engine Meeting* in San Francisco, CA, from April 15-16.
- Attended *The Tenth Annual Teaching and Learning with Technology Conference*, West Lafayette, IN, March 1 - 2, 2006.
- Attended the 2005 *Frontiers in Education Conference*, Indianapolis, IN, October 19-22, 2005.
- Attended *The Ninth Annual Teaching and Learning with Technology Conference*, West Lafayette, IN, March 1 - 2, 2005.
- Attended the *Informatics: Defining the Research Agenda Conference*, Bloomington, IN, Sep 10 – 12, 2005.
- Robert Signorelli and myself gave a presentation of our work entitled *Towards a General-Purpose Search Engine* at the 2003 Search Engine Meeting, Boston, April 2003.

Faculty Curriculum Vitae – Electrical and Computer Engineering

Name and Academic Rank:
Edward R. Doering, Associate Professor

Degrees:
Ph. D. Iowa State University, 1992
M.S. Iowa State University, 1987
B.S. Iowa State University, 1986

Rose-Hulman Service:
Years of Service at Rose-Hulman: 11 years
Original Appointment Date and Rank: September, 1994, Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s): September, 2000, Associate Professor

Professional Experience:

2000 - Present	Associate Professor ECE Department Rose-Hulman Institute of Technology, Terre Haute, IN
1994 – 2000	Assistant Professor ECE Department Rose-Hulman Institute of Technology, Terre Haute, IN
1993 - 1994	Assistant Professor EE Department Penn State University, Behrend College, Erie, PA
1992 - 1993	Assistant Professor Physics Department Bemidji State University, Bemidji, MN
1987 - 1989	Project Engineer Rosemount, Inc., Eden Prairie, MN

Consulting Activities:

2000 - 2001	Polymer Technology Systems

Patents and Disclosures:
None

Professional Registration:
None

Principal Publications of the last five years:

- Doering, E.R., G.J. Havrilla, and T.C. Miller, “Disilicide Diffusion Coating Inspection by Micro X-Ray Fluorescence Imaging,” *Journal of Nondestructive Evaluation* 23(3):95-105, 2004.
- Doering, E.R. and G.J. Havrilla, “Micro X-Ray Fluorescence Imaging for Silicide Diffusion Coating Inspection," pp. 538-545 in *Review of Progress in Quantitative NDE, Vol. 23A*, edited by D. O. Thompson and D. E. Chimenti, American Institute of Physics, New York, 2003.
- T.A. Hudson and E.R. Doering, “Custom Chips to Developed to Reinforce Learning Integrated Circuit Design,” pp. 123-125 in *Proceedings of the 2003 International Conference on Microelectronics Systems Education*, 2003.
- Hudson, T.A., E.R. Doering and G. Lee-Thomas, “Teaching Mixed-Signal Integrated Circuit Design to Undergraduates,” *IEEE Transactions on Education*, accepted for publication 2005.
- Doering, E. R., “Synthesis of Plucked-String Sounds in MATLAB,” *Computers in Education Journal* 11(4) 9-15, 2001.

Membership in Scientific and Professional Societies:

IEEE

Selected Honors and Awards:

- Helen Plants Award for “Most Innovative Workshop,” *The VRML Confabulation: An Experiment with a New Dissemination Mode* (with Mark Yoder), Frontiers in Education Conference, November 1996.

Institutional and Professional Service of the last five years:

- Chair, Laptop Committee (Rose-Hulman Institute of Technology)
- TechSmith Educational Advisory Board

Professional Development Activities of the last five years:

- *Circuits Learned by Example Online (CLEO)*, Ed Doering and Xiaoyan Mu, NSF CCLI Program, Phase I Exploratory, $124,594, 2006-08.
- NASA/ASEE Summer Faculty Fellow, Marshall Space Flight Center, Huntsville, AL (Summer 2004)
- NASA/ASEE Summer Faculty Fellow, Johnson Space Center / White Sands Test Facility, Las Cruces, NM (Summer 2002)

Faculty Curriculum Vitae - Mathematics Department

Name and Academic Rank:
S. Allen Broughton, Professor and Head of Mathematics

Degrees:

Ph.D.	1982	Mathematics	Queen's University, Ontario, Canada (1978-82)
M.Sc.	1978	Mathematics	Queen's University, Ontario, Canada (1975-78)
B.Sc.	1975	Mathematics	University of Windsor, Ontario, Canada (1971-73,74-75)

Rose-Hulman Service:

Years of Service at Rose-Hulman:	12 years
Original Appointment Date and Rank:	September, 1994, Full Professor and Department Head

Professional Experience:

1994 – Present	**Professor and Department Head** Mathematics Department Rose-Hulman Institute of Technology, Terre Haute, IN
1986 – 1994	**Assistant Professor (1986)-89 Associate Professor (1986-94)** Mathematics Department Cleveland State University
1979 – 1985	**Van Vleck Visiting Assistant Professor** Mathematics Department University of Wisconsin-Madison
1981 – 1978	**Assistant Professor** Memorial University of Newfoundland
1974 – 1975	**Sublieutenant** Canadian Armed Forces

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: **N**one

Principal Publications of the last five years:

Publications

Constructing Kaleidoscopic Tiling Polygons in the Hyperbolic Plane, American Mathematical Monthly, 107 #8 (2000), 689-710.

Divisible tilings of the hyperbolic plane, (with D.M. Haney, L. McKeough and B. Smith) New York Journal of Mathematics, 6 (2000), 237-283.
http://nyjm.albany.edu:8000/j/2000/6-12.pdf

Anharmonic Vibrational Motions in C60 : A Potential Energy Surface Derived from Vibrational Self Consistent Field Calculations, D. Jelski, Laszlo Nemes, S. Allen Broughton, Journal of Cluster Science, Vol 16, No 1, March 2005.

Membership in Scientific and Professional Societies:
1986-present member, AMS, American Mathematical Society
1986-present member, MAA, Mathematical Association of America and Indiana Section MAA
1994-present member, SIAM, Society for the Industrial Applications of Mathematics

Selected Honors and Awards: none

Institutional and Professional Service of the last five years:
Chair, Rose-Hulman Institute of Technology Curriculum Committee
Chair, Rose-Hulman Institute of Technology Academic Software Review Team
Chair, ECE Head Search Committee
Member, Laptop Computer Selection Committee
Advisor Math Double Majors and Minors

Professional Development Activities of the last five years:

Grants
Computational Group Theory and Hyperbolic Geometry, NSF-REU, DMS-9619714 (extension), $120,000, P.I.
Computational Group Theory, Hyperbolic Geometry, Number Theory, and Inverse problems NSF-REU, DMS-0097804, $144,000, P.I
2003-2004 Sophomore Course and Ancillaries in Nanoscience DMR-0304487 $100,000 senior investigator
2003-2004 MAA Conference Grant supported by NSF DMS-0241090, $2,500.

Presentations
The Unreasonable Effectiveness of Mathematics, Rose-Hulman Symposium to open 126'th school year, August 2000.
Transform Methods in Image Processing, Mathematics Faculty Seminar, Mount Holyoke College, Spring 2001. http://www.rose-hulman.edu/~brought/Epubs/mhc/mhctransimage.html
Higher Genus Soccer Balls, Mount Holyoke Math Club, Spring 2001. http://www.rose-hulman.edu/~brought/Epubs/soccer/soccer.html
Signals, Images, ..., What's Next in Scientific Visualization, Sigma Xi, Mount Holyoke College, Spring 2001. http://www.rose-hulman.edu/~brought/Epubs/sigmaxi/sciviz.html
The Rose-Hulman Laptop Program, (with Ed Doering), Ohio Northern University, September 2001.
Automorphisms of Riemann Surfaces, Galois Groups, and Hecke Algebras, Rose Math Seminar, March 2002.
Vanishing Cycles and Kaleidoscopic Quadrilateral Tilings, Rose Math Seminar, December 2002. http://www.rose-hulman.edu/~brought/Epubs/quads/quads.html
Kaleidoscopic Tilings on Surfaces, This Time with the Groups, Rose Math Seminar, Spring 2003, http://www.rose-hulman.edu/~brought/Epubs/withgroups/withgroups.html
Are the Students Competent Users of Mathematics?, AMS Meeting, Phoenix, January, 2004, http://www.rose-hulman.edu/~brought/Epubs/laptop/Phoenix.html
Equivalence of Real Elliptic Curves, Rose Math Seminar, October 2004. http://www.rose-hulman.edu/~brought/Epubs/realelliptic/realelliptic.html
Fostering Undergraduate Research in Mathematics, Showcase - Best Assessment Practices VII, Rose-Hulman, Arpil 2005. http://www.rose-hulman.edu/~brought/Epubs/showcase/foster.html
Enumeration of the Equisymmetric Strata of the Moduli Space of Surfaces of Low Genus, AMS Regional Conference, Santa Barbara, April 2005. http://www.rose-hulman.edu/~brought/Epubs/SantaBarbara/santabarbara.htm

Faculty Curriculum Vitae – Physics and Optical Engineering Department

Name and Academic Rank:
Sudipa Mitra-Kirtley, Associate Professor of Physics and Optical Engineering

Degrees:

Ph.D.	1991	Physics	University of KY, Lexington, KY
B.Sc.	1983	Physics	Berea College, KY

Rose-Hulman Service:

August 1993-Present: Associate Professor (associate position since 1998), Department of Physics and Applied Optics, Rose-Hulman Institute of Technology, Terre Haute, IN

April 1991-August 1993: Joint post-doctoral scientist at Lawrence Berkeley Laboratory, CA, and Schlumberger-Doll Research, CT.

Professional Experience:

April 1991-August 1993: Joint post-doctoral scientist at Lawrence Berkeley Laboratory, CA, and Schlumberger-Doll Research, CT.

Consulting Activities:

Schlumberger-Doll Research, CT

Principal Publications of the last five years:

1. S. Mitra-Kirtley, "Bringing Physics Lectures Outside of Classroom", Proceedings of International Conference on Physics Education, India, 2005.
2. "New Interactive Software is an A+ Instructional Tool", By NewsDesk, Converge Online, a case-study on use of DyKnow by S. Mitra-Kirtley.
3. Roger Wiltfong, Sudipa Mitra-Kirtley*, Oliver C. Mullins, Ballard Andrews, Go Fujisawa, and John W. Larsen, Sulfur speciation in different kerogens by XANES spectroscopy, Energy and Fuels, Sept 2005..
4. S. Mitra-Kirtley, O.C. Mullins, Sulfur Chemical Moieties in Carbonaceous Materials, book chapter through final revisions, O. C. Mullins, E. Sheu M. Hamami, and A. Marshall, eds., Kluwer Academic Press.
5. Sudipa Mitra-Kirtley, and Oliver C. Mullins, Sulfur species in asphaltene, resin, and oil fractions of crude oils by XANES and IR spectroscopy methods, National APS meeting Proceedings, March, 2002, and 2001 ASTM Conference Proceedings, July, 2001.

Principal Presentations of the last five years:

1. "Wireless technology in physics classes", International Conference on Physics Education, New Delhi, India, August, 2005.
2. Was invited to a workshop (all expenses paid by NSF) on Activity Based Physics, Univ. of OR, July, 2005.
3. "Physics with HP tablets" at the Best Assessment Symposium, RHIT, April, 2005.
4. "DyKnow in Physics classrooms" Teaching Enhancement series, RHIT, Feb. 2005.
5. "Best Practices with DyKnow software", DyKnow, Indianapolis. January 2005.
6. "Use of HP tablets in physics classes and laboratories", to delegates from Kanazawa Institute of Technology. December 2004.
7. Conducted a workshop, "Use of HP tablets in physics classes and laboratories", to delegates from Kanazawa Institute of Technology. December 2004
8. "The World is our Classroom", at the Hewlett-Packard Higher Education for Technology Teaching Conference, Monterey, CA. November 2004.
9. Presented a talk on Sulfur Speciation in different kerogens by XANES spectroscopy, *National Synchrotron Light Source seminar series*, Brookhaven National Lab., April, 2004.
10. "Sulfur speciation on different kerogens", poster presentation, Applied Statistical Physics and Molecular Engineering Annual Conference, Puerto Vallarta, Mexico, August, 2003.
11. "Wireless Computers Enhance Education Methods", at "No Strings Attached" conference, Case-Western University, April,

2003.

12. "Sulfur species in asphaltene, resin, and oil fractions of crude oils by XANES and IR spectroscopy methods", Applied Statistical Physics and Molecular Engineering Annual Conference, Cancun, Mexico, July, 2001.

Membership in Scientific and Professional Societies:

Member-Sigma Xi

Selected Honors and Awards:

Baccalaureate Constituency Group Representative, National Sigma Xi
Member, Sigma Pi Sigma, a physics honor society
Dean's honor list student, Berea College
Outstanding service award, Rose-Hulman Institute of Technology, 2005

Institutional and Professional Service of the last five years:

Faculty advisor – Physics undergraduates and Optical Engineering Masters students (3).
Member – Faculty Affairs Committee.
Member-Academic Quality and Instructional Planning
Member-Honors and Awards committee
Introduced Studio Physics method of teaching at Rose-Hulman Institute of Technology, 1995.
First to use DyKnow with tablet PC's in classrooms at Rose-Hulman Institute of Technology
Served on executive committee, Sigma Xi chapter of Wabash Valley.
Served as president, Sigma Xi, Wabash Valley chapter.

Professional Development Activities of the last five years:

1. Submitted a proposal to Weaver Grant Award, "A proposal for the study of the different sulfur structures in limestone-derived kerogens and crude oils by XANES and EXAFS techniques, April, 2005.
2. S. Mitra-Kirtley submitted a proposal, "REU Site: Research Experience on Spectroscopy and Microscopy: XANES at BNL, and SRD, SEM, and UV-NIR Fluorescence at RHIT," to NSF in the amount of $205,276, June, 2004.
3. Submitted numerous proposals to Brookhaven National Laboratory, beamtime at National Synchrotron Light Source, for research beamtime with undergraduate students, after joining Rose-Hulman: March, 2006, February 2005, June, 2005, April 2004, September, 2003, July 2003, August 2002, April, 2002., May- August, 2002, May-August, 1996, November, 1995, May-August, 1995, September-December, 1994, and May-August, 1994. The proposals were based on different x-ray absorption research areas. (All of the above beamtimes were funded).
4. Submitted an Adaptation and Implementation proposal to the NSF titled "Redesigned physics at Rose-Hulman Institute of Technology", in the amount of $127,344, December, 2004.
5. Submitted a proposal to Brookhaven National Laboratory (BNL), Faculty-Student Research Assistance Program, for support of travel and other expenses of a RHIT student to perform experiments at BNL, April, 2004. (Funded)
6. Submitted a proposal to Brookhaven National Laboratory (BNL), Faculty-Student Research Assistance Program, for support of travel and other expenses of a RHIT student to perform experiments at BNL, February, 2003. (Funded)
7. Co-authored two proposals to HP Company. The awards were of the amounts $220,500 in 2003, and $130,500 in 2004. (Funded)
8. Submitted a CCLI-NSF proposal, with co-investigator, Dr. Maarij Syed, named "Interactive Physics incorporating wireless technology for beginning Science and Engineering students" June, 2003, for $24,000.
9. Submitted a proposal, "Proposal for three students' directed research on fluorescence studies of crude oil samples" for Weaver Award, RHIT, Spring, 2003.
10. Submitted a proposal to NSF in the amount of $400,000, to be used as scholarship funds to assist 31 students for four years, February, 2003, with several other faculty members from RHIT.
11. Requested equipment (spectrophotometer) to study the fluorescence of crude oils, a $25,000 value, December, 2002. (Funded)
12. Submitted a CCLI proposal in the amount of $149,722 for "Redesigned Physics at Rose-Hulman Institute of Technology", December, 2002, with Dr. Maarij Syed from RHIT.
13. Several mini-grants received from Schlumberger-Doll Research to carry out research with fluorescence spectroscopy at Rose-Hulman Institute of Technology, August, 2001.(Funded)

Faculty Curriculum Vitae – Physics and Optical Engineering Department

Name and Academic Rank:
Charles Joenathan, Professor of Physics and Optical Engineering

Degrees:

Ph.D.	1986	Physics and Optics	Indian Institute of Technology, Madras, India
M.S.	1980	Physics	American College, Madurai, India
B.Sc.	1978	Physics	American College, Madurai, India

Rose-Hulman Service:

Years of Service at Rose-Hulman:	14 years
Original Appointment Date and Rank:	September, 1991, Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s):	September, 1997 Associate: September, 2001, Full Professor

Professional Experience:

Sept. 91 – Present	**Asst. Professor/Assoc. Professor/Professor, (Department Head since 1999)** Department of Physics and Optical Engineering,
July 96 - Aug. 97	**Visiting Professor**, Mechanical Engineering, University of Stuttgart, Germany
June 93 - July 93	**Visiting Research Scientist**, Alexander von Humboldt Fellow, Institute for Applied Optics, Mechanical Engineering, University of Stuttgart, Germany
Nov. 89 - Aug. 91	**Research Associate**, Center for Applied Optics Studies Rose-Hulman Institute of Technology, Terre Haute, IN 47803, USA
Jul. 88 - Oct. 89	**Visiting Research Scientist**, Alexander von Humboldt Fellow, Institute for Applied Optics, Mechanical Engineering, University of Stuttgart, Germany
Jan. 86 - Feb. 88	**Senior Project Officer**, Engineering Design Center, IIT, Madras, India
Jan. 84 - Jan. 86	**Project Associate**, Engineering Design Center, IIT, Madras, India

Consulting Activities:

a. MPD Inc., Trilithic, Fiber Optics Wide Band Radar
b. Dimensional Analysis, - Germany

Patents and Disclosures:

1. Ein Verfahren zur Relativ Pruefung Optischer Komponenten Speckle - Messung (A speckle method to compare optical components) Patent submitted in Germany [1989].
2. Decorrelation of time varying laser speckles of fruits and vegetables, B. M. Khorana, C. Joenathan, and Z. Xu, submitted for disclosure on June 13, [1994].
3. A grating electronic speckle shearing pattern interferometer, C. Joenathan and L. Buerkle, Submitted for disclosure in August [1994].
4. A holographic element collimation tester, C. Joenathan, Submitted for disclosure, October [1994].
5. Dual magnification telescope, Submitted for disclosure, June [2000].
6. Speckle method to obtain biological activity of tissues, Patent number: 20040152989 , June [2002].

Professional Registration: None

Principal Publications of the last five years:

1. C. Joenathan and H.J. Tizani, "Speckle Metrology", Book Chapter, VCH publication in Optics Encyclopedia, Germany, 2004.
2. Optical Engineering at Rose-Hulman Institute of Technology: ABET Accreditation Steps Taken With EC 2000, Charles Joenathan, Robert Bunch, and Azad Siahmakoun, ETOP, [2003].

3. Outcome based education in the Optics curriculum for over 10 years at Rose-Hulman, to be communicated to OPN [2003].
4. On the beam ratio in temporal speckle pattern interferometers, C. Joenathan, Communicated to Optics and Laser Technology [2005]
5. Optical fiber diameter measurement with a simplified Lloyds mirror system, C. Joenathan, T. Clevenger, and R. Bunch, To be Communicated to Applied Optics (2006)

Membership in Scientific and Professional Societies:

Fellow - Optical Society of America
Fellow - SPIE, the international society for optical engineers
Fellow - Alexander Humboldt fellow association of America
Fellow - Optical Society of India,
Member – American Society of Engineering Educators

Selected Honors and Awards:

Young Scientist of India for the year 1987, Award presented by President of India
Humboldt fellow to carry out independent research in Germany (1988 &1989),
Fellow of the Optical Society of India (June 1995)
Cited by the Marques Who's Who in Science and Engineering in USA
Recipient of the Rose-Hulman's Outstanding Scholar Award 1996-97
Fellow of the Optical Society of America (OSA) 1997
Fellow of the Society of Optical Engineering (SPIE) 2006

Institutional and Professional Service of the last five years:

Faculty advisor – Optical Engineering majors
Member - Curriculum committee
2000 Roundtable discussion on Master's education
2001 Program committee, SPIE conference in Germany
2002 APS roundtable conference on physics education
2003 Program committee, SPIE conference
2004 Program committee, Regional SPIE conference, Philadelphia
1. Initiated the exchange programs with University of Applied Sciences, Ulm, Germany
2. Initiated the exchange programs with University of Applied Sciences, Jena, Germany
3. First step to initiate exchange programs with IIT, Delhi, Madras, and Mumbai, India, April 2002.

Professional Development Activities of the last five years:

1. C. Joenathan, Physics Educational panel discussion, Franklin Institute, Philadelphia, November, 2002
2. C. Joenathan, R.M. Bunch and Z. Milanovic, "Optical engineering foundation curriculum development with 'optics tool kit'," submitted proposal to NSF, $75,000. June 2003
3. C. Joenathan, R. M. Bunch, and Z. Milanovic, "ABET accreditation of optical engineering at Rose-Hulman: Incorporating EC2000 criteria in a science department", OSA Education and Training in Optics and Photonics, Tuscan, AZ, 2003
4. C. Joenathan, "Temporal Speckles and its application for object shape and deformation measurement", Annual meeting of the Indiana Academy of Science a, Anderson University 2003.
5. C. Joenathan, R. Bunch, and S. Granieri, Accreditation of an Optical Engineering Program – Setting up Educational Objectives and Outcomes for a New Program, Best Assessment Process, March, 1 (2006)

Faculty Curriculum Vitae - Chemistry Department

Name and Academic Rank:
Rebecca B. DeVasher, Professor of Chemistry

Degrees:

Ph.D.	2004	Organometallic Chemistry	The University of Alabama
B.S.	2000	Chemistry	The University of Alabama

Rose-Hulman Service:

Years of Service at Rose-Hulman:	2 years
Original Appointment Date and Rank:	September, 2004, Visiting Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s):	September, 2005, Assistant Professor

Professional Experience:

2004 – Present	**Visiting Assistant Professor/Assistant Professor** Chemistry Department Rose-Hulman Institute of Technology, Terre Haute, IN
2003-2004	**Instructor** The University of Alabama, Tuscaloosa, AL
2000-2004	**Research Assistant** The University of Alabama, Tuscaloosa, AL
2001-2004	**Supplemental Instruction Leader and Department Liaison** The University of Alabama, Tuscaloosa, AL
2000-2001	**Teaching Assistant** The University of Alabama, Tuscaloosa, AL

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: None.

Principal Publications of the last five years:

Shaughnessy, K. H.; Booth, R. S. "Sterically Demanding, Water-Soluble Alkylphosphines as Ligands for High Activity Suzuki Coupling in Aqueous Solvents," *Org. Lett*, **2001**, *3*, 2757-2759.

DeVasher, R. B.; Moore, L. R.; Shaughnessy, K. H. "Aqueous-Phase, Palladium-Catalyzed Cross-Coupling of Aryl Bromides Under Mild Conditions Using Water-Soluble, Sterically Demanding Alkylphosphines," *J. Org. Chem.*, **2004,** 69(23), pp. 7919-7927.

DeVasher, R. B.; Spruell, J. M.; Dixon, D. A.; Broker, G. A.; Rogers, R. D. "Experimental and Computational Study of Steric and Electronic Effects on the Coordination of Bulky, Water-Soluble Alkylphosphines to Palladium under Reducing Conditions: Correlation to Catalytic Activity," *Organometallics*, **2005,** 24(5), pp. 962-971.

Shaughnessy, K. H.; DeVasher, R. B. "Palladium-Catalyzed Cross-Coupling Reactions in Aqueous Media: Recent Progress and Current Applications," *Current Organic Chemistry*, **2005**, 9, pp 1-20.

Membership in Scientific and Professional Societies:
American Chemical Society (ACS)
Alpha Chi Sigma Fraternity

Selected Honors and Awards:
Department of Energy Scholarship, 2002
The University of Alabama National Alumni Association Fellowship, 2003 – 2004

Institutional and Professional Service of the last five years:
Acted as Freshman Advisor, 2005-2006
Served on the Committee on the Assessment of Student Outcomes, 2005-2006
Served as Secretary on the Traffic and Safety Committee, 2005-2006

Professional Development Activities of the last five years:
Let student research activities, and received the Rose-Hulman Institute of Technology, Weaver Research Award, and sponsored student presentations:
Vincent Franco, Application of Amberlite® IRA-410 CL Resin as a Base in Environmentally Friendly Palladium-Catalyzed Cross-Coupling Reactions, *17th Annual Butler University Undergraduate Research Conference*, Indianapolis, IN (April, 2005)
Polland, Ross, Environmentally Friendly Synthesis of 4'Hydroxyacetophenone, *18th Annual Butler University Undergraduate Research Conference*, Indianapolis, IN (April, 2006)

Faculty Curriculum Vitae – Humanities and Social Sciences Department

Name and Academic Rank:

Caroline Carvill, Professor of American Literature

Degrees:

Ph.D.	1989	American Literature	University of Arkansas
Masters	1983	English	University of Arkansas
B.S.	1980	English	University of Arkansas

Rose-Hulman Service:

Years of Service at Rose-Hulman:	17 years
Original Appointment Date and Rank:	September, 1989, Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s):	September, 1994, Associate Professor
	September, 2002, Full Professor

Professional Experience:

2002-Present **Professor (Department Head since 2004)**
Humanities and Social Sciences Department
Rose-Hulman Institute of Technology, Terre Haute, IN

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: None

Principal Publications of the last five years:

"'You might as well listen to the chain gang:' *The Ballad of the Sad Café*, forthcoming in *Reflections of Critical Eye: Essays on Carson McCullers*," University Press of American, 2006.

Invited Chapter, "Feminist and Gender Criticism," in A Companion to William Faulkner Studies, Robert Hamblin and Charles Peek, eds., Greenwood Press,2003.

"The Dixie Association: Where Jefferson Davis and Martin Luther King Intersect," in Baseball/Literature/Culture: Essays, 1995-2001, Peter Carino, ed., McFarland & Company, Inc., September 2003.

"Light in August" and "The Sound and the Fury," in Cyclopedia of Literary Places, R. Baird Shuman, Ed., Salem Press, 2003.

"Miller Williams," in Critical Survey of Poetry, 2nd Revised Edition, Tracy Iron-Georges, ed., Salem Press, 2002.

"Integrating Writing into Technical Courses" (With Julia Williams, Susan Smith, Anneliese Watt), Proceedings of the 2002 ASEE Conference.

"The Gift Outright," in Masterplots II: Poetry, Revised Edition, Tracy Iron-Georges, ed., Salem Press, March 2002.

Membership in Scientific and Professional Societies:

American Society for Engineering Education
Center for the Study of Southern Culture
National Council of Teachers of English
Modern Language Association
National Society for Experiential Education

Selected Honors and Awards:

Rose-Hulman Alumni Association Honorary Alumnus Award	2003
Chair, Liberal Education Division, American Society for Engineering Education (ASEE)	2001-2002
Who's Who Among America's Teachers	2002, 2006

Institutional and Professional Service of the last five years:

Advisory Committee for the Selection of a New President	2005-2006
Freshman Advisor	2001-2006
Diversity Commission	2001-2006
Institute Curriculum Committee	2003-
Chair, Rules and Discipline Committee	2003-2004
Peer Reviewer, ASEE Liberal Education Division	2001-2004
Computer Use Policy	2002-2003
Chair, Advisory Committee on Selection of New VPAA/Dean	2001-2002

Professional Development Activities of the last five years:

"Liberal Education and Assessing for ABET," American Society for Engineering Education Conference, Chicago, June 2006.

"Service Learning at Rose-Hulman Institute of Technology," American Society for Engineering Education Conference, Salt Lake City, Utah, June 2004.

"Opportunities and Rewards for Service Learning in Engineering," International Conference on Civic Education Research, New Orleans, LA, November 2003.

"Engaging Students in the Classroom," American Society for Engineering Education Conference, Nashville, TN, June 2003.

"Faculty-Agency Collaboration in Service Learning" (with Carrie McKillip), National Society for Experiential Education Conference, Las Vegas, NV, October 2002.

"Integrating Writing into Technical Courses" (With Julia Williams, Susan Smith, Anneliese Watt), ASEE Conference, Montreal, Canada, June 2002.

"Recruiting Faculty to Service Learning," National Society for Experiential Education (NSEE) Conference, Orlando, FL, October 2001.

Invited Panelist, "Service Learning in Engineering," American Society for Engineering Education (ASEE) Sectional Conference, Purdue University, March 2001.

Attended National Academic Chairpersons' Conference, February, 2005.

Attended Faulkner & Yoknapatawpha Conference, July 2001, 2002, 2003, 2004, 2005.

Faculty Curriculum Vitae – Humanities and Social Sciences Department

Name and Academic Rank:
Julia M. Williams, Associate Professor of English and Executive Director of the Office of Institutional Research, Planning, and Assessment

Degrees:

Ph.D.	1992	English Literature	Emory University
Masters	1990	English Literature	Emory University
Diploma	1985	Anglo-Irish Literature	Trinity College, Dublin, Ireland
B.A.	1984	English and Music	Trinity University

Rose-Hulman Service:

Years of Service at Rose-Hulman:	12 years
Original Appointment Date and Rank:	September, 1994, Assistant Professor
Date(s) of Promotion(s) and Resulting Rank(s):	September, 2000, Associate Professor

Professional Experience:

1992 – Present	**Adjunct Professor/Assistant Professor/Associate Professor** Humanities and Social Sciences Department Rose-Hulman Institute of Technology, Terre Haute, IN
1991 – 1992	**Instructor** English Department University of Tennessee, Knoxville, TN

Consulting Activities: None.

Patents and Disclosures: None.

Professional Registration: None

Principal Publications of the last five years:

"Linking Classroom and Institute Assessment of Engineering Students' Writing: The Case of Rose-Hulman Institute of Technology," *Developing Better Writers and Thinkers Through Assessment*, Stylus Press.

"How Dumb Do You Want Me to Write It?': Engineering Students, Technical Communication, and Civic Responsibility," under consideration by *Journal of Business and Technical Communication* (revise and resubmit)

"Teaching Enron: The Rhetoric and Ethics of Whistleblowing," co-author with Richard House and Anneliese Watt, IEEE *Transactions on Professional Communication: Special Issue on Case Studies for Teaching Technical and Professional Communication* 47.4 (December 2004): 244-55.

"Introduction to the Special Issue on New Case Studies for Technical and Professional Communication," IEEE *Transactions on Professional Communication: Special Issue on Case Studies for Teaching Technical and Professional Communication*, 47.4 (December 2004): 229-32.

"Technological Paternalism," Last Word column, *ASEE Prism* December 6, 2004: 72.

"The Engineering Portfolio: Communication, Reflection, and Student Learning Outcomes Assessment," *International Journal of Engineering Education* 18.2: 199-207.

"Technical Communication, Engineering, and ABET's Engineering Criteria 2000: What Lies Ahead?" *Technical Communication* 49.1 (February 2002): 89-95.

"Writing to Design/Designing to Write: Using the Correlation between Communication and Engineering to Improve Student Reflection," co-author with Jeff Froyd and Anne Watt, *Proceedings of the ASEE 2002 Conference*

"Incorporating Writing Assignments in Technical Courses," co-author with Caroline Carvill, Susan Smith, and Anne Watt, *Proceedings of the ASEE 2002 Conference*

“”How Dumb Do You Want Me to Write It’: Instilling a Sense of Civic Responsibility in Engineering Students through Technical Communication,” *Proceedings of the ASEE 2002 Conference*

Membership in Scientific and Professional Societies:
IEEE Professional Communication Society
American Society for Engineering Education
Association of Teachers of Technical Writing
Modern Language Association
Conference on College Composition and Communication

Selected Honors and Awards:
Outstanding Scholarship Award, Humanities and Social Sciences Department, Rose-Hulman Institute of Technology, 2004
Triangle Fraternity Teacher of the Year Award, Rose-Hulman Institute of Technology, November 1997
Grant for Travel to International Meetings Abroad, American Council of Learned Societies, April 1995
The Graduate School Award for Excellence in Graduate Teaching in the Humanities, Emory University, May 1991
Dissertation Fellowship, English Department, Emory University, 1990-1991
Summer Travel Grant, English Department, Emory University, 1990
Distinguished Pass, Ph.D. Oral Examination, English Department, Emory University, 1989
Teaching Assistantship, English Department, Emory University, 1988-1990
Graduate Assistantship, English Department, Emory University, 1987-1988
Graduate Fellowship, English Department, Emory University, 1986-1987
Pass with Highest Distinction, Diploma in Anglo-Irish Literature, Trinity College, Dublin, Ireland, 1985

Institutional and Professional Service of the last five years:
Interviewer, Technical Communication Position Search, Humanities and Social Sciences Department (HSS)
Events Committee, HSS
Outcomes Committee, HSS
Co-chair, Commission on the Assessment of Student Outcomes
Chair, Honors and Award Committee
Chair, Philosophy Search Committee, HSS
Editor, Challenge-X: Crossover to Sustainable Mobility Proposal
Co-author, HP Second Request Proposal
Redesign and development of RosE Portfolio system, Summer 2004
Institute Leaves Committee
Computer Science and Software Engineering Curriculum Committee
Jack Kent Cooke Foundation Scholarship Representative
Chair, Visiting Scholar Committee, HSS
Global Studies Initiative Committee, HSS
Search Committee, Computer Science Chair
Review of Institutional Objectives Team (RIOT)

Professional Development Activities of the last five years:
Editor, Tutorial Column, IEEE Transactions on Professional Communication
Reviewer, Association of Teachers of Technical Writing proposals
Reviewer, American Society for Engineering Education papers
Reviewer, Transactions of the IEEE Professional Communication Society
Reviewer, IEEE Publications
Reviewer, Harcourt Publishers
Program Chair, American Society for Engineering Education, Liberal Education Division
Chair, EngiComm Special Interest Group at College Composition and Communication Conference
Member of the Advisory Committee, IEEE PCS
Presentations and workshops at ASEE and IEEE International Professional Communication Conference

Appendix III – Other Information

A. Software Engineering Program Assessment Plan 2005-06

Software Engineering

Program Assessment Plan

2005-06 Academic Year

Coordinator

Don Bagert, Director of Software Engineering

Program Educational Objectives

Software engineering graduates will

PEO1. develop complex systems (including analysis, design, construction, maintenance, quality assurance and project management) using the appropriate theory, principles, tools and processes.
PEO2. use appropriate computer science and mathematics principles in the development of software systems.
PEO3. solve problems in a team environment through effective use of written and oral communication skills.
PEO4. have knowledge of current issues presently involved in effectively performing duties as a software practitioner in an ethical and professional manner for the benefit of society.
PEO5. practice the lifelong learning needed in order to keep current as new issues emerge.
PEO6. develop software in at least one application domain.

Description of Assessment Tools for Program Educational Objectives

- Alumni Survey Results – A survey of all alumni of the program, conducted by the Institute, with specific questions asked to graduates of a particular major.
- Board of Advisors Minutes – The meeting minutes of the CSSE Board of Advisors, which meets once or twice a year.

Description of Assessment of Program Educational Objectives

Each program objective (shown above) will be assessed using both assessment tools (Alumni Survey Results and Board of Advisors minutes).

Success Criteria for Program Educational Objectives

- Success on the alumni survey for a particular Program Educational Objective is that at least 60% of the software engineering alumni who graduated 3-5 years ago that respond feel that they have met that objective
- Success on a Board of Advisors report means that the board has expressed satisfaction on that report for the Program Educational Objectives.

Program Outcomes

By graduation, software engineering graduates will have demonstrated

SE1. the ability to apply software engineering theory, principles, tools and processes, as well as the theory and principles of computer science and mathematics, to the development and maintenance of complex software systems.
SE2. the ability to design and experiment with software prototypes.
SE3. the ability to select and use software metrics.
SE4. the ability to participate productively on software project teams involving students from both software engineering and other majors.
SE5. effective communication skills through oral and written reports and software documentation evaluated by both peers and faculty.
SE6. the ability to elicit, analyze and specify software requirements through a productive working relationship with project stakeholders.
SE7. the ability to evaluate the business and impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues.
SE8. the ability to apply appropriate codes of ethics and professional conduct to the solution of software engineering problems.
SE9. the knowledge required to understand the need for and the ability to perform in lifelong learning.
SE10. the basic knowledge required in a software engineering application domain track.

Description of Assessment Tools for Program Outcomes

Course Assessment Reports – The assessment reports of all Computer Science and Software Engineering (CSSE) courses that can be taken by software engineering majors.

Exit Interviews – Each graduate of the software engineering program is interviewed one-on-one by the Director of Software Engineering shortly before the last day of classes of their last term at Rose-Hulman.

Application Domain Curricular Requirements – These are the minimal requirements for an application domain track, which are that 1) it must be related to a significant software application area, and 2) it must contain at least 12 hours worth of courses taught outside of the CSSE department.

Description of Assessment of Program Outcomes

Each of Program Outcomes SE1 through SE10 will be assessed using the Exit Interviews.

Each of Program Outcomes SE1 through SE9 will also be assessed using the Course Assessment Reports. Program Outcome SE10 is also assessed through Application Domain Curricular Requirements.

Success Criteria for Program Outcomes

- Success on an exit interview means that 80% or more of those interviewed expressed satisfaction when asked about a particular program outcome.
- For the course assessment reports, success for a particular program outcome is defined as success for all assessments made for each course related to that particular outcome.
- Success for Application Domain Curricular Requirements means that 100% of software engineering graduates have met those requirements.

Terminology

“Develop software” means to analyze, design, implement, verify, test and maintain software systems.

Appendix III

B. Software Engineering Program Assessment Report 2005-06

Software Engineering

Program Assessment Report

2005-06 Academic Year

Coordinator

Don Bagert, Director of Software Engineering

Program Educational Objectives

Software engineering graduates will

PEO1. develop complex systems (including analysis, design, construction, maintenance, quality assurance and project management) using the appropriate theory, principles, tools and processes.
PEO2. use appropriate computer science and mathematics principles in the development of software systems.
PEO3. solve problems in a team environment through effective use of written and oral communication skills.
PEO4. have knowledge of current issues presently involved in effectively performing duties as a software practitioner in an ethical and professional manner for the benefit of society.
PEO5. practice the lifelong learning needed in order to keep current as new issues emerge.
PEO6. develop software in at least one application domain.

Description of Assessment Tools for Program Educational Objectives

- Alumni Survey Results – A survey of all alumni of the program, conducted by the Institute, with specific questions asked to graduates of a particular major.
- Board of Advisors Minutes – The meeting minutes of the CSSE Board of Advisors, which meets once or twice a year.

Description of Assessment of Program Educational Objectives

Each program objective (shown above) will be assessed using both assessment tools (Alumni Survey Results and Board of Advisors minutes).

Success Criteria for Program Educational Objectives

- Success on the alumni survey for a particular Program Educational Objective is that at least 60% of the software engineering alumni who graduated 3-5 years ago that respond feel that they have met that objective
- Success on a Board of Advisors report means that the board has expressed satisfaction on that report for the Program Educational Objectives.

Objective Evaluation of Program Educational Objectives

The success criterion holds for the Board of Advisors. The alumni survey cannot be done until 2008-09, since the first software engineering program graduates were in 2005.

Coordinator's Subjective Evaluation of Program Educational Objectives

I am satisfied with the Program Educational Objectives so far.

Recommendations for Next Year for Program Educational Objectives

No changes are recommended.

Program Outcomes

By graduation, software engineering graduates will have demonstrated

SE1. the ability to apply software engineering theory, principles, tools and processes, as well as the theory and principles of computer science and mathematics, to the development and maintenance of complex software systems.
SE2. the ability to design and experiment with software prototypes.
SE3. the ability to select and use software metrics.
SE4. the ability to participate productively on software project teams involving students from both software engineering and other majors.
SE5. effective communication skills through oral and written reports and software documentation evaluated by both peers and faculty.
SE6. the ability to elicit, analyze and specify software requirements through a productive working relationship with project stakeholders.
SE7. the ability to evaluate the business and impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues.
SE8. the ability to apply appropriate codes of ethics and professional conduct to the solution of software engineering problems.
SE9. the knowledge required to understand the need for and the ability to perform in lifelong learning.
SE10. the basic knowledge required in a software engineering application domain track.

Description of Assessment Tools for Program Outcomes

Course Assessment Reports – The assessment reports of all Computer Science and Software Engineering (CSSE) courses that can be taken by software engineering majors.

Exit Interviews – Each graduate of the software engineering program is interviewed one-on-one by the Director of Software Engineering shortly before the last day of classes of their last term at Rose-Hulman.

Application Domain Curricular Requirements – These are the minimal requirements for an application domain track, which are that 1) it must be related to a significant software application area, and 2) it must contain at least 12 hours worth of courses taught outside of the CSSE department.

Description of Assessment of Program Outcomes

Each of Program Outcomes SE1 through SE10 will be assessed using the Exit Interviews.

Each of Program Outcomes SE1 through SE9 will also be assessed using the Course Assessment Reports. Program Outcome SE10 is also assessed through Application Domain Curricular Requirements.

Success Criteria for Program Outcomes

- Success on an exit interview means that 80% or more of those interviewed expressed satisfaction when asked about a particular program outcome.
- For the course assessment reports, success for a particular program outcome is defined as success for all assessments made for each course related to that particular outcome.
- Success for Application Domain Curricular Requirements means that 100% of software engineering graduates have met those requirements.

Objective Evaluation of Program Outcomes

The success criterion holds for all tool-outcome pairs, with the following exceptions:

- CSSE 220 Course Assessment Report (fall) – 3 outcomes did not hold
- CSSE 230 Course Assessment Report (spring) – 1 outcome did not hold
- CSSE 332 Course Assessment Report (winter) – 1 outcome did not hold
- CSSE 332 Course Assessment Report (spring) – 1 outcome did not hold

This meant that, according to the success criteria, SE1, SE4 and SE5 did not hold. However, those Program Outcomes were successful in most cases:

- SE1 held for 20 of the 22 Course Assessment Reports involving it (90.9%),
- SE4 held for 9 out of the 11 Course Assessment Reports involving it (81.8%), and
- SE5 held for 4 out of the 6 Course Assessment Reports involving it (66.7%).

Coordinator's Subjective Evaluation of Program Outcomes

I am satisfied with the Program Outcomes so far, since even those Outcomes that failed were successful in a supermajority of cases.

Recommendations for Next Year for Program Outcomes

- I recommend no changes in the program outcomes for next year.
- The bar appears to be too high for the course assessment reports, and it needs to be re-evaluated.

Terminology

“Develop software” means to analyze, design, implement, verify, test and maintain software systems.

Appendix III

C. Sample Course Assessment Plan

CSSE 376 Software Quality Assurance

Course Assessment Plan

Spring 2006

Instructor

Mark Ardis (one section)

Learning Objectives

Students who complete this course should be able to:

1. Perform formal reviews of software artifacts
2. Create a test plan for a software system
3. Apply different strategies for unit-level and system-level testing
4. Understand principles and strategies of integration and regression testing
5. Use a problem tracking system
6. Evaluate a user interface for suitability
7. Understand purposes of quality processes, methods for measuring that quality, and standards used

Description of Assessment Tools

Daily Quizzes - short (5-10 questions) quizzes completed during class

Biweekly Homework Assignments – problems related to software engineering and software project management.

Project - team project in which the students plan and conduct usability testing of software artifacts.

Exams – two one-hour exams.

Course Assessment Matrix

	Objective						
	1	2	3	4	5	6	7
Quizzes	X				X		
Homework		X	X			X	
Project	X	X	X	X	X	X	X
Exams				X			X

Success Criteria

The course will be considered fully successful if the following statement holds for every tool-objective pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning objective is in the proficient range.

Appendix III

D. Sample Course Assessment Report

CSSE 376 Software Quality Assurance

Course Assessment Report

Spring 2006

Instructor

Mark Ardis

Recommendations from Previous Offering and Corresponding Changes

- (none)

Other Significant Changes from Previous Offering

- (none)

Learning Outcomes

Students who complete this course should be able to:

1. Perform formal reviews of software artifacts
2. Create a test plan for a software system
3. Apply different strategies for unit-level and system-level testing
4. Understand principles and strategies of integration and regression testing
5. Use a problem tracking system
6. Evaluate a user interface for suitability
7. Understand purposes of quality processes, methods for measuring that quality, and standards used

Description of Assessment Tools

Daily Quizzes - short (5-10 questions) quizzes completed during class

Biweekly Homework Assignments – problems related to software engineering and software project management.

Project - team project in which the students plan and conduct usability testing of software artifacts.

Exams – two one-hour exams.

Course Assessment Matrix

	Outcome						
	1	2	3	4	5	6	7
Quizzes	X				X		
Homework		X	X			X	
Project	X	X	X	X	X	X	X
Exams				X			X

Success Criteria

The course will be considered fully successful if the following statement holds for every tool-outcome pair selected above:

- Among the students who earn proficient grades in the course, the average grade on the portions of the assessment tools that are relevant to the learning outcome is in the proficient range.

Objective Evaluation

The success criterion holds for all tool-outcome pairs.

Summary of Relevant Student Feedback

- Students like the test-first approach, Dilbert cartoons, and the usability testing unit.
- Some students find this course to be a little slow.

Instructors' Subjective Evaluation

- I am satisfied with the students' learning and experience in the course.

Recommendations for Next Offering

- Find a better textbook for the basic testing material (Culbertson et al.) or just drop it and use lecture notes.
- Change the assignments so each week there is one individual assignment and one team assignment. That would give students more experience with some of the testing methods, and it would make it easier for them to do the project work.
- Add more coverage of automated testing.
- Find a more challenging project than RoseyCalendar?.
- As more testing is taught in 120-220-230 some of the topics in this course can be dropped.