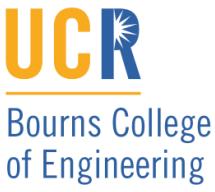
****

**University of California Riverside**

**Bourns College of Engineering**

**Computer Science**

July 1, 2012

Submitted to the

Computer Accreditation Commission

Accreditation Board for Engineering and Technology, Inc.

111 Market Place, Suite 1050

Baltimore, Maryland 21202-401

Table of Contents

[CRITERION 1. STUDENTS 14](#_Toc324338807)

[B.1 Enforcing Prerequisites 17](#_Toc324338808)

[Diversity in the Bourns College of Engineering 30](#_Toc324338809)

[CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES 31](#_Toc324338810)

[CRITERION 3. STUDENT OUTCOMES 43](#_Toc324338811)

[CRITERION 4. CONTINUOUS IMPROVEMENT 48](#_Toc324338812)

[CRITERION 5. CURRICULUM 84](#_Toc324338813)

[CRITERION 6. FACULTY 100](#_Toc324338814)

[Table 6-1. Faculty Qualifications 111](#_Toc324338815)

[Table 6-2. Faculty Workload Summary 113](#_Toc324338817)

[CRITERION 7. FACILITIES 115](#_Toc324338818)

[CRITERION 8. INSTITUTIONAL SUPPORT 123](#_Toc324338819)

[B. Program Budget and Financial Support 123](#_Toc324338820)

[C. Staffing 124](#_Toc324338821)

[E. Support of Faculty Professional Development 126](#_Toc324338822)

[PROGRAM CRITERIA 127](#_Toc324338823)

[Appendix A – Course Syllabi 128](#_Toc324338824)

[Appendix B – Faculty Vitae 129](#_Toc324338825)

[Appendix C – Equipment 130](#_Toc324338826)

[Appendix D – Institutional Summary 131](#_Toc324338827)

[1. The Institution 131](#_Toc324338828)

[Table D-1. Program Enrollment and Degree Data 135](#_Toc324338829)

[Signature Attesting to Compliance 137](#_Toc324338830)

**ABET**

**Self-Study Report**

**for the**

**Computer Science Program**

**at**

**University of California Riverside**

**Riverside**

July 1, 2012

**CONFIDENTIAL**

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

**BACKGROUND INFORMATION**

1. **Contact Information**

|  |  |
| --- | --- |
| http://static.cs.ucr.edu/store/images/sized/7da6e22b132161f9f6d717390d9821bf.jpg | **Dr. Laxmi N. Bhuyan**  Chair of Department of Computer Science and Engineering  Department of Computer Science and Engineering  Winston Chung Hall, Room 351  University of California Riverside  Riverside, CA 92521  Email: *bhuyan@cs.ucr.edu*  Phone +1 951 827-2244  Fax +1 951 827 4643  Office Room 421 |

Please CC all email contacts to Dr. Bhuyan to

* Stefano Lonardi <stelo@cs.ucr.edu>
* Dennis Rice <drice@engr.ucr.edu>
* Mitch Boretz <mitch.boretz@ucr.edu>
* Eamonn Keogh <eamonn@cs.ucr.edu>

All requests for additional information will be acknowledged within 24 hours.

Note to evaluators: We have mostly used a consecutive numbering scheme for the tables in this document. I.e. the tables are numbered Table 1, Table 2 etc. However we realize that ABET evaluators are used to considering certain canonical tables, such as *Table 5-1 Curriculum* and *Table 6-1. Faculty Qualifications* etc. For these handful of ABET canonical tables we use the standard ABET numbering system.

1. **Program History**

We begin with a brief timeline of significant events in the Computer Science Department’s history, shown in Table 1.

Table 1: A Brief Timeline of Significant Events in the Programs History

|  |  |
| --- | --- |
| 1972: CS track added to BS in math.  1972: BS and MS in Applied Science approved, both with CS tracks.  1976: Applied Science shut down and BS in CS established within Math Dept.  1977: First graduates from BS in CS.  1982: MS in CS started in Math Dept.  1984: First graduates from MS in CS.  1984: Math Dept changes name to Department of Mathematics and Computer Science.  1988: CoE proposal approved. Glenn Everett appointed as Acting Dean  1989: First BS students admitted to CoE.  1990: Math and CS become separate departments within CNAS  1990: Larry Larmore becomes chair of new CS Department  1990: College of Engineering hires Susan Hackwood as founding dean.  1991: CS moves from Sproul Hall to University Office Building.  1992: CS joins CoE as its first “department” --- others are still “programs.”  1992: Teodor Przymusinski becomes chair  1994: PhD in CS approved.  1994: Tom Payne becomes CS chair | 1995: CoE becomes BCOE, Bourns Hall opens, CS moves in along with rest of BCOE.  1995: Susan Hackwood resigns deanship and Lee Rudi (retired UCSD dean of engineering) becomes interim dean  1997: Satish Tripathi becomes BCOE dean.  1998: First CS PhD (Torsten Berger) graduated.  1998: Dept of Computer Science renamed Department of Computer Science and Engineering.  1999: Mart Molle becomes CS chair.  2001: BS in Computer Engineering established (jointly administered by CS and EE)  2001: CSE moves to Surge Building.  2002: Tom Payne begins second term as chair.  2003: Mark Matsumoto becomes Acting Dean of Engineering.  2005: Reza Abbaschian becomes dean of BCOE  2005: CSE and EE move to Engineering II (now renamed Winston Chung Hall).  2007: BS in Computer Science and BS in Computer Engineering both receive ABET accreditation.  2007: BS/MS in CS approved.  2007: Laxmi Bhuyan becomes chair of CSE.  2010: Computer Engineering (which is hosted by both the CS and EE departments) becomes an official interdisciplinary program with its own committee-in-charge. |

From its inception in 1976, UCR’s BS degree in Computer Science was envisioned to involve a balance of both hardware and software and of both theory and practice. It was housed in the Department of Mathematics which taught both the theoretical and the software components of the program, including required courses introduction to programming (Pascal); assembly language; algorithms and data structures; architecture; compilers; operating systems; automata, formal languages, and computability; plus electives in switching theory; information theory; and modeling and simulation. The hardware portion of the program was taught by the Physics Department and included digital electronics and the design, construction and programming of microprocessor-based systems.

In 1982, an MS in CS was established. Y.C. Hong was hired in 1983. He improved the offering in architecture and established an elective in data bases. Marek Chrobak was hired in 1986 and strengthened the offerings in data structures and algorithms, and in automata, formal languages, and computability.

In 1989, Lawrence Larmore was hired, further strengthening the offerings in those areas.

In 1990, the Department of Mathematics and Computer Science split, amicably, into two separate departments, which remained within the College of Natural and Agricultural Sciences, and the Campus Computing Center took over administration of both departments’ computing facilities.

In 1991, Yu-Chin Hsu, whose area was design automation (for digital systems) was hired and instituted a more design-oriented approach to architecture.

Also, Teodor Przymusinski was hired and instituted elective offering in artificial intelligence.

In 1992, per prior plans, CS became part of the College of Engineering and structured laboratories were instituted for all undergraduate courses except those in theory of computation. Also, at that point the courses in digital electronics and in microprocessor-based design were moved from Physics to Electrical Engineering.

In 1993, the Department got funding to establish its own computing facilities and decided to standardize on Linux as the OS for its desktop facilities. At roughly the same time the Department switched from Pascal to C++ as the main programming language for its undergraduate offerings.

In 1994, a Ph.D. in CS was begun, which made support of PhD students a significant consideration in the awarding of TAships. At that point the faculty grew rapidly and the breadth and depth of the undergraduate offerings grew accordingly. Among the resulting innovations were:

* A required two quarter sequence in logic design and embedded systems jointly listed and taught with Electrical Engineering, plus additional electives in design automation and embedded systems.
* A required three course introductory sequence in programming and data structures.
* Electives in graphics, computational geometry, and video games.
* Electives in data mining and machine learning were added.
* The AI offering became more statistics (machine learning) oriented.
* An elective in software engineering was added.
* Electives in networking and in security were added.
* Separate degrees in Computer Engineering and in Information System (later changed to Business Informatics) were established.

Table 2 shows the CSE enrollment and number of degrees awarded since the 1992 academic year.

**Table 2: Computer Science Enrollment and Degrees**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **New Students** | | **Total Enrollment** | |  | **Degrees Granted** | |  | |  | **Freshmen** | **Transfer** | **UG** | **Grad** | **Total** | **UG** | **Grad** | **Total** | | 1992-93 | 34 | 18 | 112 |  | 112 | 18 | 6 | 24 | | 1993-94 | 27 | 19 | 130 | 29 | 159 | 9 | 7 | 16 | | 1994-95 | 57 | 21 | 190 | 38 | 228 | 13 | 10 | 23 | | 1995-96 | 39 | 34 | 232 | 38 | 270 | 24 | 9 | 33 | | 1996-97 | 66 | 27 | 243 | 37 | 280 | 21 | 12 | 33 | | 1997-98 | 140 | 52 | 392 | 46 | 438 | 26 | 12 | 38 | | 1998-99 | 116 | 60 | 463 | 41 | 504 | 46 | 14 | 60 | | 1999-00 | 172 | 83 | 595 | 52 | 647 | 53 | 8 | 61 | | 2000-01 | 153 | 87 | 700 | 79 | 779 | 72 | 16 | 88 | | 2001-02 | 195 | 88 | 850 | 88 | 938 | 111 | 16 | 127 | | 2002-03 | 140 | 43 | 867 | 103 | 970 | 143 | 17 | 160 | | 2003-04 | 97 | 32 | 709 | 131 | 840 | 174 | 25 | 199 | | 2004-05 | 67 | 19 | 463 | 141 | 604 | 120 | 33 | 153 | | 2005-06 | 50 | 11 | 328 | 124 | 452 | 118 | 41 | 159 | | 2006-07 | 74 | 8 | 264 | 106 | 370 | 63 | 33 | 96 | | 2007-08 | 79 | 6 | 257 | 117 | 374 | 57 | 38 | 95 | | 2008-09 | 92 | 8 | 273 | 137 | 410 | 42 | 29 | 71 | | 2009-10 | 82 | 5 | 252 | 152 | 404 | 9 | 35 | 44 | | 2010-11 | 124 | 6 | 304 | 143 | 447 | 37 | 26 | 63 | | 2011-12 | 89 | 7 | 320 | 156 | 476 |  |  |  | |

1. **Options**

The program has no options, tracks or concentrations at this time.

1. **Organizational Structure**

Table 3 shows the administrative structure of the Computer Science program.

Table 3: The Organizational Structure of CE

|  |
| --- |
|  |

1. **Program Delivery Modes**

All courses for the bachelor’s degree are delivered in campus classrooms and laboratories on weekdays and weeknights. The curriculum includes no cooperative education, distance education, or web-based instruction.

1. **Program Locations**

All courses are delivered on the campus of the University of California, Riverside.

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

The CS program did have a weakness with regard to *coverage of social and ethical issues in computing*. Dr. Tom Payne wrote a “Due Process” response on 21st of February 2007 and ABET then responded with a clean bill of health in July 2007. The detailed documentation for this is in “*Response to CAC’s Draft Statement: Department of Computer Science, UC Riverside February 21, 2007*” available for inspection. In brief, we addressed this issue by:

* Augmenting one of the CS program’s outcomes, namely outcome J, which previously read: “*a knowledge of contemporary issues*,” to now read: “*a knowledge of contemporary issues, including ethical and social issues*.”
* Making outcome J an *explicit* student outcome of four upper-division (i.e. junior/senior level) required courses:
  + – CS 152 (Compiler Design)
  + – CS 153 (Design of Operating Systems)
  + – CS 161 (Design and Architecture of Computer Systems)
  + – CS 179 (Project in Computer Science)
* Modified the syllabi of the four above courses to explicitly note the aspects of “*ethical and social issues*” to be covered in each case.
* Prepared an instructor’s manual with a taxonomy of social and ethical topics to cover.

**Important Note:** The sole CS weaknesses from our last evaluation is addressed above. However the Computer Engineering program (CEN) also had two weaknesses. Because the weaknesses involved the Program Educational Objectives, and because the Computer Science program shares many courses with Computer Engineering, any changes to Computer Engineering must necessarily greatly affect Computer Science. Therefore, for completeness we discuss below how the CEN weaknesses were very carefully addressed, with full consultation with both the Computer Science program and the Electrical Engineering program.

The CE weaknesses are addressed in great detail in a 109-page document. This document was sent to ABET on June 20th 2008. Below we briefly review the two issues, and how we resolved them; the details are in “CE-response-062008.pdf”, sent to ABET in 2008, and also available on request.

We had both a Criteria 2 and 3 Weaknesses:

**Weaknesses 1**: Criterion 2. Program Educational Objectives.

Criterion 2 states, “*... program educational objectives are broad statements that describe the career and professional accomplishments that the program is preparing graduates to achieve.”* The evaluators findings stated: the program’s objectives are not broad statements that describe the accomplishments of computer engineering graduates and their achievements; instead they describe skills more appropriately articulated in program outcomes. In addition, it is not clear that these objectives were reached based on the needs of program constituents (students, faculty, employers, advisory boards, and the community at large). Since these objectives were not defined based on the needs of program constituents, it is not clear how the results are used to improve program outcomes and for graduates to attain the objectives.

**Weaknesses 2**: Criterion 3. Program Outcomes and Assessment

Criterion 3 states, “*There must be processes to produce these outcomes and an assessment process, with documented results, that demonstrates that these program outcomes are being measured and indicates the degree to which the outcomes are achieved. There must be evidence that the results of this assessment process are applied to the further development of the program*.” The evaluators stated: Course objectives are defined for each course but they are not clearly related to program outcomes that are referred to as departmental outcomes. It is stated in the report that the college will administer a new assessment tool in the fall of 2006 but the process used presently in measurement of program out comes is not documented. Achievement of program outcomes is demonstrated using course objectives and grades in homework assignments and exams. Sufficient evidence was not provided to demonstrate students attain the outcomes articulated by the computer engineering program.

We note that the department *fully acknowledged* both weaknesses, and worked tirelessly to address them.

We addressed **Weakness 1** by the following:

In late October 2007, the two faculty members most responsible for ABET, Dr. Eamonn Keogh and Dr. Tom Payne, had a series of meetings with the interested parties, including the CE Assessment and Accreditation committee, the EE Assessment and Accreditation committee (Dr. Amit K. Roy-Chowdhury and Dr Roger Lake), the CE Undergrad education committee and the Chair of CSE, Dr. Laxmi Bhuyan.

On November 6th 2007, Dr. Eamonn Keogh and Dr. Tom Payne wrote new PEOs. They based them very closely on the EE PEO. The Electrical Engineering Department was consulted and asked for feedback at this stage.

On November 8th 2007, Dr. Eamonn Keogh presented the new PEOs to the CSE Board of Advisors, from 1:00pm to 1:30pm. Each member got a take home copy, and was invited to discuss the PEOs both at the meeting, and offline by email at a later date.

On November 14th 2007, Dr. Eamonn Keogh presented the new PEOs with notes from the Board of Advisors to the entire CSE faculty, 40 minutes were spent discussing the PEOs

and they were adopted by a majority vote.

Dr. Eamonn Keogh solicited feedback and approval for our new PEOs from employers of our CE students. This process was conducted by phone, email and in four cases, by an onsite visit by a delegation from our department. We also solicited comment and approval for our new PEOs from our students.

In summary, we created new PEOs after soliciting input and approval from all our constituents, our Board of Advisors, employers of our students, faculty, and the students themselves. Furthermore, we have carefully documented this process at every step. This detailed documentation can be found in “*CE-response-062008.pdf*”.

We addressed **Weakness 2** by the following:

To remedy the Criterion-3 weakness, UCR’s Computer Engineering program (CEN) adopted the outcomes-assessment process of UCR’s Electrical Engineering program (EE). This was done for the following reasons:

1. CE is jointly administered by the Department of Electrical Engineering and by the Department of Computer Science and Engineering (CSE).

2. The engineering portion of CE’s curriculum is composed of EE courses, taught by the Electrical Engineering Department, and CS courses, taught by CSE.

3. The CE and EE programs have the same outcomes, namely ABET’s A-K outcomes.

4. Following EE’s due-process response, EAC’s Final Statement of 2007 judged EE’s outcomes-assessment process to fulfill ABET’s Criterion-3 requirements, and indeed extensively complimented them.

The complication of joint administration by two departments is handled as follows. Both departments follow the same (EE’s) outcomes-assessment process with analysis and evaluation being performed by their respective Assessment Committees. Recommendations go to their respective faculties. Changes to courses are handled within the corresponding department following standard interdepartmental consultation. Changes to the curriculum must be approved by both departments. In principle, a change could be approved by one department and denied by the other. That has *never* happened, and it will be up to the dean to modify the organizational structure if and when such a problem arises.

The detailed documentation for all the above can be found in “*CE-response-062008.pdf*”.

1. **Joint Accreditation**

This program is seeking CAC accreditation only. Note that we are having a simultaneous EAC evaluation of our Computer Engineering program.

|  |  |
| --- | --- |
| CS 010: Introduction to Comp Sci for Science, Math, and Engineering I:  **Key**: Where possible, we have placed the names of courses *inline* in this document. However, in most of the tables, and some of the text, this is unwieldy. Thus we have produced this key. We suggest you print this page separately and use it for reference.  CS 011: Introduction to Discrete Structures:  CS 012: Introduction to Comp Sci for Science, Math, and Engineering II  CS 013: Introductory Computer Science for Engineering Majors:  CS 014: Introduction to Data Structures and Algorithms:  CS 021: Introduction to UNIX:  CS 030: Introduction to Computational Science and Engineering:  CS 049 (E-Z): Language Laboratory:  CS 049E: Introductory C and C++:  CS 049G: Advanced C++:  CS 049I: C#:  CS 049J: Introductory Java:  CS 049M: Matlab:  CS 049N: Hardware Description:  CS 049Q: Perl:  CS 049S: Bash:  CS 049Y: Python:  CS 061: Machine Organization and Assembly Language Programming:  CS 066: Introduction to Three-Dimensional Digital Modeling:  CS 067: Three-Dimensional Digital Modeling and Animation:  CS 100: Software Construction:  CS 111: Discrete Structures:  CS 120A: Logic Design:  CS 120B: Introduction to Embedded Systems:  CS 122A: Intermediate Embedded and Real-Time Systems:  CS 122B: Advanced Embedded and Real-Time Systems:  CS 130: Computer Graphics:  CS 133: Computational Geometry:  CS 134: Video Game Creation and Design:  CS 141: Intermediate Data Structures and Algorithms:  CS 143: Multimedia Technologies and Programming:  CS 145: Combinatorial Optimization Algorithms:  CS 150: The Theory of Automata and Formal Languages:  CS 151: Introduction to Theory of Computation:  CS 152: Compiler Design:  CS 153: Design of Operating Systems:  CS 160: Concurrent Programming and Parallel Systems:  CS 161L: Laboratory in Design and Architecture of Computer Systems:  CS 162: Computer Architecture:  CS 164: Computer Networks:  CS 165: Computer Security:  CS 166: Database Management Systems:  CS 168: Introduction to Very Large Scale Integration (VLSI) Design:  CS 169: Mobile Wireless Networks:  CS 170: Introduction to Artificial Intelligence:  CS 171: Introduction to Expert Systems: (*Will be removed by 2013*)  CS 177: Modeling and Simulation:  CS 179(E-Z): Project in Computer Science:  CS 179E: Compilers:  CS 179F: Operating Systems:  CS 179G: Database Systems:  CS 179I: Networks:  CS 179J: Computer Architecture and Embedded Systems:  CS 179K: Software Engineering:  CS 179M: Artificial Intelligence:  CS 179N: Graphics and Electronic Games:  CS 180: Introduction to Software Engineering:  CS 181: Principles of Programming Languages:  CS 183: UNIX System Administration:  CS 190: Special Studies:  CS 193: Design Project:  CS 194: Independent Reading:  CS 198I: Individual Internship in Computer Science: | EE 140: Computer Visualization:  ENGR 001G Professional Development & Mentoring  ENGR 101G Professional Development & Mentoring  ENGR 180W Technical Communications  ENGL 001A Beginning Composition  ENGL 001B Intermediate Composition  ENGL 01SC Applied Intermediate Composition for Science  and Engineering Majors  MATH 009A First-Year Calculus  MATH 009B First-Year Calculus  MATH 009C First-Year Calculus  MATH 010A Multivariable Calculus  MATH 046 Introduction to Ordinary Differential Equations  MATH 113 Linear Algebra  MATH 120 Optimization  MATH 126 Introduction to Combinatorics  MATH 135A Numerical Analysis  MATH 135B Numerical Analysis  PHYS 040A General Physics  PHYS 040B General Physics  PHYS 040C General Physics  STAT 155 Probability & Statistics for Engineering  PHIL 008 Introduction to Logic  PHIL 124 Formal Logic |
| **Abbreviations**    BCOE: Bourns College of Engineering  CS: Computer Science  CSE: Computer Science and Engineering (department)  CS&E: Computer Science and Engineering (department), Alterative abbreviation, rarely used. Does not appear in this document, but may appear in old self studies and related texts.  CEN: UC wide abbreviation for Computer Engineering Programs. |

**GENERAL CRITERIA**

# CRITERION 1. STUDENTS

1. **Student Admissions**

The admissions processes for all our engineering degree programs conform to the UCR Academic Senate’s interpretation of the admission policies of the University of California, which, in turn, interpret the mandates of the California Master Plan for Higher Education.

In broad terms, the Master Plan constrains the University of California to admitting only students ranking in the top 12.5% of the high school graduates in the State. Students in lower tiers are eligible for admission to campuses of the California State University system, or to community colleges. Placement in the top 12.5% of the graduating class is determined by the UC Eligibility Index, which is computed centrally by the UC Office of the President, based on criteria defined by the UC System-Wide Academic Senate.

Figure 1 summarizes the freshman admissions process to our college. Prospective students submit their applications to the Office of Admissions for the University of California, which serves all ten campuses. Applicants may apply to multiple campuses, and to multiple programs at these campuses. They may also designate primary and alternate majors. The UC Office of Admissions determines whether each applicant meets the UC Eligibility criteria (which specify GPA and coursework requirements) and forwards each eligible application to the campuses to which admission is being sought. Ineligible applicants are rejected.

If a student is UC-eligible but is not selected for admission to the campus(es) that he or she applied to, admission to another UC campus is offered. It is notable that the Riverside campus switched from a referral campus to a selective campus within the past four years. That is, because of the increasing number and quality of students applying directly to UCR, we no longer offer admission to students who are UC-eligible but declined by their first-choice campuses. Nevertheless, we remain the most diverse campus of the UC system (in terms of overall numbers; on a percentage basis, UC Merced has greater diversity because of its very small student population), with a substantial number of students who are the first in their families to attend college.

Within UCR, processing of these freshman applications begins through the Campus Office of Admissions, in accordance with guidelines defined by the Undergraduate Admissions Committee (UAC) of the UCR Academic Senate. An Enrollment Management Council (EMC) also exists at the campus level to make decisions annually on the enrollment targets at the campus and college levels. These decisions are informed by the strategic planning processes at the campus and College levels.

UCR follows a multi-tier admissions process, which operates as follows. At the first tier, an Academic Index Score (AIS) is computed for each applicant, based primarily on academic parameters such as the Grade-Point-Average (GPA), the Scholastic Aptitude Test (SAT) score, and the number of completed Advanced Placement or International Baccalaureate (IB) courses. College-specific upper and lower AIS thresholds are determined in accordance with the planned enrollment targets. All applicants to a college whose AIS scores exceed the upper threshold are automatically admitted to their program of interest. All applicants with AIS scores below the lower threshold for each college are removed from that college’s pool. The remaining applicants are forwarded to the respective colleges for further processing.

Once these forwarded applications arrive at BCOE, a, BCOE-specific Index Score (BIS) is computed for each applicant. This BIS score is a function of the applicant’s grades in Mathematics and Science, as well as the math part of the SAT Reasoning Test (the SAT Advanced test is not required by UC). The applicants to each program are ranked by BIS score, and applicants are admitted starting at the top of the list for each program until the program’s enrollment target is met. Applicants may be placed on a wait list, to be admitted if the yield rate from the admitted pool is insufficient to satisfy program targets.

|  |
| --- |
|  |

Figure 1: The admissions process begins with an application to the UC system, which is forwarded to the campus and then to the college for consideration.

1. **Evaluating Student Performance**

Student performance monitoring is primarily the role of the Office of Student Affairs, under the supervision of the Associate Dean for Undergraduate Affairs, Professor C.V. Ravishankar. Each program also has a faculty member designated as the Program Faculty Adviser, who serves as the primary departmental contact for program-specific policy decisions. College-level policy is under the purview of the Associate Dean. The staff of the Office of Student Affairs (OSA) supports the undergraduate programs.

Each student is assigned to a staff adviser in the OSA, and encouraged to meet with this adviser whenever the need arises, but at least once per quarter. In addition, attendance at a mandatory Annual Major Advising session is required of all undergraduates in the college. The Annual Major Advising session is conducted jointly by the OSA staff and the Program’s Faculty Adviser, and provides information on a variety of topics to students, including program requirements as well as academic success strategies and professional development opportunities.

Figure 2 depicts the process for monitoring student progress. Students are required to maintain a GPA of 2.0[[1]](#footnote-1) each quarter, as well as cumulatively. Students are reminded of these requirements regularly, first during the registration process in their first quarter as freshmen, and again each year during Annual Major Advising. Grades are posted by instructors each quarter to the central Student Information System (SIS) database, which tracks student performance, and provides degree audits to check for completion of degree requirements. At the end of each quarter, staff advisers in the OSA review the academic records of BCOE students and identify all whose term and cumulative GPAs are below 2.0.

A failure to meet these GPA requirements results in a student being placed on probation. The student is notified of this probationary status, and advised that a failure to obtain at least a 2.0 GPA the following term will result in dismissal. A registration hold is now placed on the student’s record at that point, to be released only upon the completion of Academic Success Workshops and other advising and mentoring activities through the OSA. A student who receives a dismissal notice may appeal the dismissal to the Associate Dean, who may grant or reject the appeal based on extenuating circumstances.

The primary source of information regarding student performance is the campus-wide Student Information System (SIS) (which is maintained by the campus Computing and Communications office), that records all student registrations and grades, and which is maintained by the Computing and Communications organization. All staff and faculty advisers have access to this system, either directly, or through the Student Advising System (SAS) front-end that provides access to student transcripts and degree audits. The staff of the OSA uses this system regularly to monitor student progress.

Students who are about to graduate are required to complete a graduation application. At this point, the student’s academic adviser in OSA performs a detailed manual check to ensure that all degree requirements have been met. If the requirements have been met, the Office of the Registrar is notified of degree completion, so that the degree may be awarded.

# B.1 Enforcing Prerequisites

All students are given a term-by-term course plan that ensures timely graduation as long as courses are completed in a timely manner. This course plan incorporates prerequisites, so that students who follow the course plan automatically satisfy prerequisites.

Whether or not students follow this course plan, prerequisites are enforced by the registration system. Students register for courses through the GROWL[[2]](#footnote-2) system that interfaces with SIS, and is able to enforce prerequisites. A student prevented from taking a course due to lack of prerequisites can petition the course instructor, who has the authority to grant the student a prerequisite waiver. The student is not permitted to take the course without such a waiver. Such waivers are generally approved for outstanding students, transfer students and in very special situations.

|  |
| --- |
|  |

Figure 2: Academic Advising and Performance Monitoring

1. **Transfer Students and Transfer Courses**

Transfer students apply using the same application portal that freshmen use. This portal is maintained by the System-Wide Office of Admissions, located in Oakland, CA. This office collects applications and forwards them to the UCR Office of Admissions.

In accordance with the California Master Plan for Higher Education, the University of California maintains extensive articulation agreements with Community Colleges in the State. Course articulations are reviewed and approved by the cognizant departments, and are tracked and maintained by the Campus Articulation Officer. All system-wide articulation agreements are available at the website *www.assist.org*, which is open access. The transfer route appears to be gaining popularity, especially given recent increases in tuition. When a transfer applicant (typically, from out of state) presents a transcript containing courses that have not already been articulated, the staff of the BCOE OSA collect the relevant course syllabi and work with the cognizant departments at UCR to determine articulations.

All BCOE programs have published detailed requirements for transfer admission. Admission to our programs requires a minimum GPA of 2.8, and the completion of coursework specific to the major being applied to. Incoming transfer students may transfer up to 105 quarter units (70 semester units) towards their degrees from the University. To ease the burden of consulting *www.assist.org* for each major an applicant may be interested in, we have prepared brochures showing transfer requirements for each of our majors. We make these brochures available both in hardcopy, as well as on the Web. Some examples appear at *www.engr.ucr.edu/undergrads/transferring/SpecialAgreements.html*.

If the transfer applicant for a major meets all the requirements specified by that major, the UCR Office of Admissions admits that applicant. Applicants who satisfy most transfer requirements are forwarded to the College for additional review. The OSA staff reviews these applications, and in consultation with the departments and the Associate Dean, grants exceptions as warranted. Conditional admission is also sometimes granted, subject to the completion of some requirements that may not have been met at the time of application. Below we explicitly list the UCR Transfer Admission Criteria:

**UCR Transfer Admission Criteria:**

* Complete 60 transferable units (90 quarter units) with a minimum GPA of 2.4 for California residents and 2.8 for nonresidents
* Complete (with a grade of C or better) the following course pattern:
* Two transferable college courses (3 semester or 4-5 quarter units) in English composition
* One transferable college course (3 semester or 4-5 quarter units) in mathematical concepts and quantitative reasoning
* Four transferable college courses (3 semester or 4-5 quarter units) chosen from two of the following subject areas: arts & humanities; social & behavioral science; physical & biological sciences.

Given these, students must also meet General BCOE Transfer Admission Requirements:

**General BCOE Transfer Admission Requirements:**

* A cumulative GPA of at least 2.8.
* Completion of 2 major-specific sequences for your intended major with a minimum 2.50 GPA. One sequence must be single-variable calculus (MATH 9A, 9B, 9C). The second sequence may be a sequence such as PHYS 40A, 40B, 40C.
* Completion of one year of college level English Composition (ENGL1A, 1B, 1C).

Students considering transferring to CS are counseled that in addition to the general requirements listed above, students applying for transfer admission to BCOE must also complete major-specific course work as listed below.

**Computer Science Transfer Admission Requirements:**

The following courses must be completed at the time of application:

• one course in computer programming (CS 10)

• one course in object oriented programming (CS 12)

• one course in calculus based physics with lab (PHYS 40A)

A minimum of THREE (3) additional courses (shown below) must also be completed in order to form a coherent sequence. A list of potential sequences for this major is listed below.

• two courses in calculus based physics with labs (PHYS 40B, 40C)

• one course in data structures (CS 14)

• one course in machine organization and assembly language programming (CS 61)

• one course in introduction to discrete structures (CS/Math 11)

• one course in calculus of several variables I (Math 10A)

Potential Course Sequences for Computer Science: Three courses from CS 10, 11, 12, 14, and 61 or PHYS 40A, 40B, and 40C.

These transfer rules may appear somewhat intimidating to students, but fortunately, as we discuss in the next section, we have an excellent advising program and staff.

**Advising and Career Guidance**

The Office of Student Academic Affairs (OSAA) implements and enforces academic policies developed by UCR/BCOE & its Departments/programs. There is constant consultation and feedback between faculty and academic advisors. Below we review the mission of OSAA.

MISSION: The Office of Student Academic Affairs mission is to support engineering students in achieving their educational goals by providing guidance and services which enhance their academic development. We strive to fulfill this mission by:

* *Upholding academic policies of the university, BCOE and its departments.*
* *Assisting students in acclimating to and navigating the academic environment, policies and expectations.*
* *Working intentionally to build respect, trust and cooperation with students in support of their academic success.*
* *Considering individual student needs while encouraging student development.*
* *Encouraging academic planning, self-awareness, accountability and resourcefulness.*
* *Helping students respond proactively and productively to issues impacting academic success.*
* *Committing to excellence, the academic counseling profession and continued development.*

In Table 4 we list the current OSAA staff, with brief biographical details. Note that they have decades of combined experience, and that we have an exceptionally low turnover rate.

Table 4: The OSAA Staff with Brief Biographic Details

|  |  |  |
| --- | --- | --- |
| Rod Smith | Rod Smith | M.B.A., Business Administration, University of California Irvine, June 1994. 15 years in student affairs, 6 of those at BCOE. |
| Tara Brown | Tara Brown | Master of Science in Counseling, College Counseling/Student Affairs. California State University, Northridge, May 2002, 9 years in student affairs, 5 of those at BCOE. |
| Nikki Measor | Nikki Measor | M.S. in Higher Education and Student Affairs, Indiana University, Bloomington, May 2003. 9 years in student affairs, 2 of those at BCOE. |
| Amber Scott | Amber Scott | M.S., Counseling & Guidance (Specialization in College Student Personnel), California Lutheran University, June 2007. 10 years in student affairs, 2. 5 of those at BCOE. |
| Terri Phonharath | Terri Phonharath | B.A., Political Science/Admin Studies, UCR, June 1998. 12 years in student affairs, 5 of those at BCOE. |
| Sonia De La Torre-Iniguez | Sonia De La Torre-Iniguez | M.S., Educational Counseling and Guidance with Pupil Personnel Services Credential, CSU San Bernardino, June 2010. 9 years in student affairs, 8 of those at BCOE. |
| Thomas McGraw | Thomas McGraw | M.S., Sport Management, California Baptist University, June 2006. 14 years in student affairs, 9 of those at BCOE. |
| Jun Wang | Jun Wang | M.B.A., Business Administration, University of California Riverside, June 2007. 5 years in student professional development at BCOE |

Note that the Computer Science program recognizes the importance of the highest quality student advising and career guidance. With this in mind we conduct annual informal

|  |  |
| --- | --- |
| surveys of our *lower* division students (note that this is *in addition* to the college level surveys, discussed below). We do this because the exit survey only measures *senior* students, and thus we would have a significant time lag to correct problems in lower division advising if we *only* consider exit surveys.  Our survey is conducted by giving every lower division student a copy of the *anonymous* feedback form shown in Figure 3, and asking him/her to fill it in and return to the faculty. All such forms are then tabulated and summarized, and presented at the next faculty meeting. This takes place in the middle of the Spring quarter. The results of the 2012 survey are shown Table 5.  In addition to numerical data, the survey allows the students an opportunity for a free text narrative of issues, concerns and criticisms. | Figure 3: The anonymous survey that lower-division students are asked to complete regarding their advising |

In general, the 2012 survey overwhelming contained positive statements about advising, and praise for particular staff members. One minor criticism that appeared on two responses noted a problem with an out-of-date webpage, giving dated advice. This problem was fixed within 24 hours of the survey being collected.

Table 5: The results of the Spring 2012 Survey on Academic Advising

|  |  |  |  |
| --- | --- | --- | --- |
| Spring 2012: Total of 57 Responses | Agree | Neutral | Disagree |
| The academic advisers are very accessible | 47 | 9 | 1 |
| The academic advisers give detailed clear advice | 39 | 17 | 1 |
| The academic advisers have clearly written, useful documents/websites | 42 | 13 | 2 |

In addition to this somewhat informal survey taken in lower division classes, we also take the opportunity of the exit survey to poll the students on advising by both faculty and non-faculty. Figure 4 shows that with the 2011 numbers we compare *very favorably* with other respected institutions.

|  |
| --- |
|  |

Figure 4: The BCOE students satisfaction with advising by Non Faculty (*left*) and Faculty (*right*), contrasted with other amalgamations of institutions

Finally, in Figure 5 we show the results of a more general survey of student satisfaction with advising. Note that this is for all of BCOE majors, not just computer science majors.

|  |
| --- |
|  |

Figure 5: BCOE Advising Satisfaction Survey (May 2012)

We appear to be doing well on every question except “*Support from front desk staffing*”. This is due to the fact that this position was eliminated in the last year due to funding cuts. Associate Dean, Prof Ravishankar is currently looking into ways to address this.

**Summary for this Section:**

The computer science program strongly believes that high quality student advising is critical to attracting, retaining and serving our diverse student body. We take great care to measure how well we are serving the students, and take immediate action if problems or weaknesses are spotted.

**Mechanisms for Providing Career and Professional Guidance**

The mechanisms by which students receive academic advice have already been outlined in **Section B: Evaluating Student Performance**. Here, we will describe the mechanisms for providing Career and Professional guidance.

Professional guidance and mentoring is provided by staff (particularly, the Director of Student Professional Development), the faculty, and the Career Center. The overall College philosophy that guides all interactions with students is to ensure that they are both academically and professionally prepared to become leaders in their chosen fields. This goal is especially challenging to meet in engineering colleges.

As is typical for undergraduate programs in engineering, our students spend the first two years of their undergraduate work completing prerequisite coursework in mathematics, the sciences, and the humanities and social sciences. Unfortunately, instructors in these areas are unfamiliar with any of the engineering disciplines, and unable to motivate or mentor our students in their early years here. Consequently, our students fail to develop a clear sense of academic direction or a sense of professional pride, having no role models or mentors, either at home or on campus. Another consequence of this lack of engagement in the early years with BCOE is that it is harder for students to build effective working relationships with their peers, so they can begin to see them as technically strong, and as effective partners.

We are addressing these issues in several ways. The first of these is a series of 1-unit classes intended to promote engagement with BCOE in the early years and to help the student’s professional development in later years. This series of classes are numbered ENGR 1 (freshmen), ENGR 2 (sophomores), ENGR 101 (juniors), and ENGR 102 (seniors). These courses are intended to provide our students with involvement in Professional Development activities. Activities to be performed are program-specific, and will include projects, industry overviews and interactions, involvement with professional societies and clubs, team building, career guidance, and coverage of ethics and lifelong-learning issues. The specific list of topics in these courses includes the following:

* Participate in peer-group building activity.
* Understand Engineering as a creative process for solving real-world problems.
* Understand current and future trends in the student’s major discipline.
* Understand some analysis tools, and their use in design and practice.
* Understand the stages of development of an Engineer as a Professional
* Participate in individual and group projects.
* Participate in Professional Clubs.
* Participate in the Career Path Milestones program.
* Understand the role and importance of Ethics in the Engineering profession.
* Understand the importance of engaging in life-long learning.
* Participate in Industry visits.

These topics are presented in workshops and discussion-style activities. A suite of activities supported by the college under the Professional Development Milestones program complement the program-specific content in these courses. Examples of such activities are academically-oriented workshops on time management and study-skills, as well as professionally-oriented activities such as mock interviews, resume writing, as well as research and industrial internships. Figure 6 summarizes these milestones.

|  |
| --- |
|  |

Figure 6: Professional Development Milestones Program guides students on key activities they should be undertaking during their undergraduate years to assure that they are ready for careers or graduate school.

A total of 18 Student Professional Organizations exist in BCOE, and are supported financially by the College. These organizations are student-led, and are very active. Just over 800 students are active members of these organizations (roughly 40% of the students in College).

1. BCOE SLC (Student Leadership Council)
2. ACM (Association of Computing Machinery)
3. AIChE (American Institute of Chemical Engineers)
4. ASME (American Society of Mechanical Engineers)
5. ASQ (American Society of Quality)
6. BMES (Biomedical Engineering Society)
7. EWB (Engineers Without Border)
8. IEEE (Institute of Electrical and Electronics Engineers)
9. IEEE EDS (Electron Devices Society)
10. ION (Institute of Navigation)
11. MRS (Material Research Society)
12. NSBE (National Society of Black Engineers)
13. OSA (Optical Society of America)
14. SACNAS (Society for Advancement of Chicanos and Native Americans in Science)
15. SHPE (Society of Hispanic Professional Engineers)
16. SAE (Society of Automotive Engineers)
17. SWE (Society of Women Engineers)
18. TBP (Tau Beta Pi) – Honors Society

These organizations, under the mentorship of the Director of Student Professional Development, participate in a broad range of activities during the year. A summary appears in Table 6.

Table 6: BCOE Professional Development Events, 2011-12 Academic Year

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BCOE Professional Development Milestones Program 2,102 participants total** | | | | |  |
| **Date** | **Event** | **Students** | **Date** | **Event** | **Students** |
| 10/5/2010 | Technical Job Search Workshop | 27 | 1/19/2011 | Preparing for Engr. Technical Career Fair/Fashion Show | 72 |
| 10/11/2010 | Making Professional Connections with Western Digital | 21 | 1/20/2011 | Careers in Video Game & Animation Design | 30 |
| 10/11/2010 | Careers in Pharmaceutical Industry | 36 | 1/26/2011 | Google Info Night with Alumni | 155 |
| 10/12/2010 | Beginning Resume Writing | 15 | 1/26/2011 | Information Session with CIA | 43 |
| 10/14/2010 | Advanced Resume Writing | 17 | 1/26/2011 | Information Session with National Oilwell Varco | 44 |
| 10/18/2010 | Google Careers Info Session & Resume Workshop | 146 | 1/27/2011 | Women in STEM Careers | 37 |
| 10/19/2010 | EPA Careers Info Session & Interview Workshop | 65 | 2/9/2011 | Engineering, Science, & Metrology in Defense Industries | 54 |
| 10/19/2010 | Northrop Grumman Tech Talk | 45 | 2/15/2011 | From Internship to Career Alumni Panel | 32 |
| 10/19/2010 | CIA Information Session | 56 | 2/23/2011 | Making Professional Connections | 40 |
| 11/3/2010 | Advanced Resume Workshop with Western Digital | 24 | 3/1/2011 | Interview Skills Featuring Western Digital | 35 |
| 11/8/2010 | Careers in Sustainability | 26 | 3/2/2011 | NAVY Day at Bourns College of Engineering | 160 |
| 11/8/2010 | INROAD Mixer | 58 | 4/6/2011 | Engineering Careers in Pharmaceutical & Medicine Manufacturing | 120 |
| 11/15/2010 | Internships, What, Why & How | 40 | 4/12/2011 | Interview Skills, featuring: The Aerospace Corporation | 41 |
| 11/16/2010 | Phoenix Motorcars on Electronic Vehicles Industry | 66 | 4/12/2011 | Resumania, Featuring: Northrop Grumman | 35 |
| 11/18/2010 | Careers in Water Resources and Quality | 62 | 4/14/2011 | Coffee Chat: featuring: consolidated electrical distributors | 30 |
| 12/1/2010 | Engn Presentation Skills | 28 | 4/20/2011 | Student Intern Panel | 28 |
| 1/1/2011 | Resume Writing with Skanska Constructions | 35 | 4/21/2011 | A Day in the Life of the EPA – What we do | 48 |
| 1/10/2011 | Careers in Aviation featuring Marine Corps | 32 | 4/21/2011 | Work Green, Earn Green: Careers that save the planet | 23 |
| 1/12/2011 | UG Research Internships with NSF | 70 | 4/25/2011 | Internship: What, Why, & How? | 23 |
| **(Over one hundred and forty similar events have been deleted for brevity, full list available on request)** | | | | | |
| 5/1/2012 | Interview Skills, Featuring: Consolidated Electrical Distributors | 42 | 5/1/2012 | Yikes! I'm Graduating! | 35 |
| 5/2/2012 | Advanced Resume Writing, California Steel Industries | 29 | 5/1/2012 | Jump Start to Law School, Featuring: Kaplan | 12 |
| 5/2/2012 | Career Station |  | 5/3/2012 | Job Search Skills | 12 |

In addition, the College has a very active Undergraduate Research program. Faculty are very active participants in undergraduate research. Last year, 60 of the 83 faculty in BCOE were research mentors for undergraduates. Over 250 undergraduates worked with faculty on research projects. This research has resulted in a significant number of publications and research presentations. For example, in the 2010 Southern California Conference on Undergraduate Research, 18 of the 24 research presentations from UCR were by BCOE students. For the second year in a row, BCOE students made more presentations at SCCUR than students from any other engineering college in Southern California. See Section 6.C.1 (*An Overview of our Efforts and Achievements in Undergraduate Research*) for more details on undergraduate research.

A summary of the range of Professional Development, Mentoring, and Success program in BCOE appears in Figure 7.

|  |
| --- |
|  |

Figure 7: Professional Development, Placement, and Success Programs offered to BCOE undergraduate students.

1. **Work in Lieu of Courses**

Credit is awarded for selected International Baccalaureate Advanced Placement courses taken in high school, in accordance with the charts on pages 28—31 in the General Catalog for the University of California, Riverside.

Internships and independent study courses may not be used to satisfy College subject requirements, as per the following College regulation:

* **ENR3.2.8.** Internships and independent study courses may not be used to satisfy College subject requirements. (En 25 May 95) (Renumbered & Am 25 May 00)

Credit by Examination is awarded subject to the following College Regulations:

* **ENR2.5.1.** A student who wishes to have the privilege of examination for degree credit must be in residence and not on academic probation.
* **ENR2.5.2.** Arrangements for examination for degree credit must be made in advance with the student’s faculty adviser. The approval of the Faculty adviser, the Dean of the college, and that of the instructor who is appointed to give the examination, are necessary before the examination can be given.
* **ENR2.5.3.** The results of all examinations for degree credit are entered on the student's record in the same manner as for regular courses of instruction.

1. **Graduation Requirements**

Students must also meet campus general education requirements. See Undergraduate Studies section.

Computer Science has the following requirements for the B.S. major. CS students will follow the given course plan for their selection of courses for each quarter with the help of the CS student affair officer and the CS undergraduate advisor to ensure that they meet all the requirements before the graduation.

**CS Major Requirements**

**College requirements**

1. ENGL 01SC
2. MATH 008B or MATH 009A
3. PHYS 040A, PHYS 040B, PHYS 040C

**Lower-division requirements (60 units)**

1. ENGR 001I
2. CS 010, CS 012 or CS 013, CS 014, CS 061
3. CS 011/MATH 011
4. MATH 008B or MATH 009A, MATH 009B, MATH 009C, MATH 010A
5. PHYS 040A, PHYS 040B, PHYS 040C
6. One course of 4 or more units in an engineering discipline outside the field of computer science to be selected in consultation with a faculty advisor. (Either a lower-division or an upper-division course may be used to satisfy this requirement.)
7. ENGL 01SC

**Upper-division requirements (89 units minimum)**

1. ENGR 101I
2. CS 100, CS 141, CS 150, CS 152, CS 153, CS 161, CS 161L, CS 179 (E-Z)
3. CS 120A/EE 120A, CS 120B/EE 120B
4. CS 111/MATH 111
5. ENGR 180
6. MATH 113
7. STAT 155
8. Two courses from MATH 046, MATH 120, MATH 126, PHIL 124
9. At least 24 units of technical electives to be chosen from an approved list of courses which currently includes CS 100, CS 122A, CS 122B, CS 130, CS 133, CS 134, CS 145, CS 151, CS 160, CS 162, CS 164, CS 165, CS 166, CS 168, CS 170, CS 177, CS 179 (E-Z) (4 units maximum), CS 180, CS 181, CS 183, CS 193 (4 units maximum), EE 140, MATH 120, MATH 135A, MATH 135B. The technical electives selected must be distinct from those used to satisfy the requirements specified above.

Students may petition for exceptions to the above degree requirements. Exceptions to Computer Science course requirements must be approved by the Computer Science and Engineering undergraduate advisor or chair.

1. **Transcripts of Recent Graduates**

The program will provide transcripts from some of the most recent graduates to the visiting team along with any needed explanation of how the transcripts are to be interpreted.

1. **Diversity in the Bourns College of Engineering**

|  |  |
| --- | --- |
| As we noted earlier, the Bourns College of Engineering is proud to be one of the most diverse engineering colleges in America. The number of domestic undergraduates from underrepresented backgrounds jumped 95.6% from the fall of 2006 to the fall of 2010 (the most recent academic year for which full data are available, see Table 7). In recognition of our efforts to recruit and retain students from diverse backgrounds to engineering, ABET awarded the Bourns College of Engineering the 2009 Claire Felbinger Award for Diversity, as shown in Figure 8. Our citation read: “*In recognition of extraordinarily successful initiatives for recruiting undergraduate and graduate students from diverse and disadvantaged backgrounds, retaining them though the bachelor's degree, and advancing them to graduate studies and careers in engineering*.” Our faculty and staff truly appreciate this recognition of their efforts by ABET. | Description: ABET award Figure 8: Associate Dean C.V. Ravishankar, left, accepts the 2009 Claire Felbinger Award from ABET President-Elect David Holger |

Table 7: The number of domestic undergraduates from underrepresented backgrounds in the Bourns College of Engineering has nearly doubled since 2006.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Fall 2006 | Fall 2007 | Fall 2008 | Fall 2009 | Fall 2010 |
| Undergraduate: % domestic underrepresented | 27% | 29% | 31% | 31% | 33% |
| Undergraduate: # domestic underrepresented | 340 | 377 | 449 | 521 | 665 |
| Undergraduate: % domestic female | 12% | 12% | 15% | 17% | 17% |
| Undergraduate: # domestic female | 151 | 156 | 222 | 291 | 348 |
| Graduate: % domestic underrepresented | 16% | 21% | 18% | 16% | 17% |
| Graduate: # domestic underrepresented | 14 | 24 | 27 | 24 | 32 |

# CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES

1. **Mission Statement**

The University of California, Riverside serves the needs and enhances the quality of life of the diverse people of California, the nation and the world through knowledge – its communication, discovery, translation, application, and preservation. The undergraduate, graduate and professional degree programs; research programs; and outreach activities develop leaders who inspire, create, and enrich California’s economic, social, cultural, and environmental future.

With its roots as a Citrus Experiment Station, UC Riverside is guided by its land grant tradition of giving back by addressing some of the most vexing problems facing society. Whether it is assuring a safe, nutritious, and affordable food supply; stimulating the human mind and soul through the humanities and arts; or finding solutions to the profound challenges in education, engineering, business, healthcare, and the environment, UC Riverside is living the promise.

The mission of the Bourns College of Engineering is to:

* Produce engineers with the educational foundation and adaptive skills to serve rapidly evolving technology industries;
* Conduct nationally recognized engineering research focused on providing a technical edge for the United States;
* Contribute to knowledge of both fundamental and applied areas of engineering;
* Provide diverse curricula that will instill in our students the imagination, talents, creativity, and skills necessary for the varied and rapidly changing requirements of modern life;
* Enable our graduates to serve in a wide variety of other fields that require leadership, teamwork, decision-making and problem-solving abilities; and
* Be a catalyst for industrial growth in Inland Southern California.

The vision of the Bourns College of Engineering is to become a nationally recognized leader in engineering research and education.

1. **Program Educational Objectives**

Before listing our program educational objectives, we will take a moment to state the vision and mission of the College of Engineering, and vision and mission of the Computer Science program, since both of these *informed* the creation of our PEOs.

The vision of the College of Engineering is to *become a nationally recognized leader in engineering research and education.*

Its mission is to:

* *Produce engineers with the educational foundation and the adaptive skills to serve rapidly evolving technology industries*.
* *Conduct nationally recognized engineering research focused on providing a technical edge for the U.S*.
* *Contribute to knowledge in both fundamental and applied areas of engineering*.
* *Provide diverse curricula that will instill our students with the imagination, talents, creativity and skills necessary for the varied and rapidly changing requirements of modern life and to enable them to serve in a wide variety of other fields that require leadership, teamwork, decision making, and problem solving abilities*.
* *Be a catalyst for industrial growth in the Inland Empire[[3]](#footnote-3)*.

The vision of the Computer Science program at UC Riverside is to provide students with the knowledge and skills needed to:

* *Pursue the two primary alternatives after graduation, which are to obtain employment in industry or pursue graduate studies*.
* *Succeed in a career involving a lifelong learning process*.
* *The curriculum is also designed to provide the breadth and the intellectual discipline required to enter professional careers in fields outside engineering such as business and law*.

This vision of the Computer Science program led us to define the following Program Educational Objectives (PEOs):

Graduates of UCR’s BS degree program in Computer Science will meet high professional, ethical, and societal goals as demonstrated by:

success in *post-graduation studies as evidenced by*:

* satisfaction with the decision to further their education
* advanced degrees earned
* professional visibility (e.g., publications, presentations, patents, inventions, awards)
* professional responsibilities (e.g. professional mentoring, professional society membership and offices, reviewing and editorial work for professional journals)

success in *a chosen profession or vocation as evidenced by*:

* career satisfaction
* promotions/raises (e.g. Management leadership positions or distinguished technical positions)
* professional visibility (e.g., publications, presentations, patents, inventions, awards)
* professional responsibilities (e.g. professional registration, professional mentoring, professional society membership and offices)
* entrepreneurial activities
* consulting activities

contributions *to society as evidenced by*:

* Leadership roles
* Public service
* Mentoring / outreach activities
* Volunteer service

These PEOs are a change from the last ABET accreditation, and were changed in response to feedback from ABET to CEN and EE about their PEOs after the 2006 site visit (note, that is *not* an error, as we explain below, the *CEN/EE* feedback also informed *CS*).

Because the CS program has many overlapping courses with the CEN program, and the CEN program has many overlapping courses with the EE program, the CSE department and EE department decided in 2007 that it would be advantageous if all three programs (CS/CE/EE) could share their PEOs if at all possible. Thus a committee was formed with representatives from all programs and a single set of PEOs for all programs was drafted.

In October 2007, Dr. Eamonn Keogh and Dr. Tom Payne had a series of meetings with all the interested parties, including the CSE Assessment and Accreditation committee, the EE Assessment and Accreditation committee (Dr. Amit K. Roy-Chowdhury and Dr Roger Lake), the CSE Undergrad education committee and the Chair of CSE, Dr. Laxmi Bhuyan. On November 6th 2007, the committee finalized the new PEOs.

On November 8th 2007, Dr. Keogh presented the new PEOs to the CS Board of Advisors, from 1:00pm to 1:30pm. Each member got a take home copy, and was invited to discuss the PEOs both at the meeting, and offline by email at a later date.

On November 14th 2007, Dr. Keogh presented the new PEOs with notes from the Board of Advisors to the entire CSE faculty, 40 minutes were spent discussing the PEOs and they were adopted by a majority vote.

Dr. Keogh solicited feedback and approval for our new PEOs from employers of our CE students. This process was conducted by phone, email and in four cases, by an onsite visit by a delegation from our department. We also solicited comment and approval for our new PEOs from our students.

In summary, we created new PEOs after soliciting input and approval from all our constituents, our Board of Advisors, employers of our students, faculty, and the students themselves. Furthermore, we have carefully documented this process at every step.

Finally, the PEOs (and the documentation of the process to create them) were approved by ABET EAC for CE and EE in 2007, however this is the first time ABET is seeing these PEOS for CS.

The above mission, vision and program educational objectives are published in the college catalog and are available online at the following URL:

*http://www1.cs.ucr.edu/education/undergraduate/csmajor/*

The Computer Science and Engineering Department consult regularly with its constituencies (see Section **D. Program Constituencies**), particularly its advisory board, to review their Program Educational Objectives and update them as appropriate.

Naturally, the University and College of Engineering missions are much broader and more general than the Computer Science PEOs. However, we note that all are directed toward preparing our students to make an impact in their professional careers and all share the vision of developing leaders in industry, government, academia and society. Moreover, the PEOs articulate elements of the Computer Science curriculum that will enable our graduates to apply their knowledge, to communicate effectively, and to exercise creativity through problem-solving and to prepare our graduates for a variety of careers in industry, and academia.

1. **Consistency of the Program Educational Objectives with the Mission of the Institution**

Table 8 below illustrates the correspondence of the institutional objectives to the CS objectives. This mapping is shown to the entire faculty at least once a year (most recently on March 5th 2012) who are invited to openly discuss it and offer revisions.

Table 8: Program Educational Objectives and the Mission of the Institution

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UCR CE Mission  UCR CE PEOs  (abridged) | ..educational foundation and the adaptive skills to serve rapidly evolving technology industries. | ..nationally recognized engineering research ... | Contribute knowledge fundamental and applied areas of engineering. | instill .. skills ..to enable .. a wide variety of other fields that requires leadership, teamwork, decision making, and problem solving abilities. | Be a catalyst for industrial growth in the Inland Empire. |
| success in *post-graduation studies* |  |  |  |  |  |
| success in *a chosen profession or vocation* |  |  |  |  | \* |
| *contributions to society* |  |  |  |  |  |

\* We note that an unusually large fraction of our students are from the Inland Empire, and choose to stay in the area after graduation. For example ESRI in Redlands (the largest GIS company in the world) has hired dozens of our students, and ISCA Tech, a startup in Riverside (chemical and biological sensors) has hired six of our graduates etc.

Our program objectives are designed to produce graduates who will be well educated in the fundamental concepts of computer engineering and mathematical principle. Moreover, they will have an appreciation of the need *for*, and the skills *to* be able to continue professional development throughout their lives. Due to a progressively more globalized economy interdisciplinary teaming and communication skills are becoming increasingly important, and as such we prepare graduates to function responsibly in such diverse environments.

1. **Program Constituencies**

The constituencies of the Computer Science program are the students, faculty, employers, alumni, the Advisory Board, and the community at large. The faculty has primary responsibility for educating the students and to effect the program’s educational objectives. The current students in the program and Computer Science alumni are essential constituencies. The current members of the Computer Science Advisory Board are listed in Table 9. Given that a large fraction of the Advisory Board members are associated with industry, the Advisory Board serves as an important bridge to our graduates’ employer constituency.

Table 9: Computer Science Department Board of Advisors

|  |  |
| --- | --- |
| **Name** | **Affiliation** |
| Dr. Amit Agrawal | Auryn, Inc |
| Mr. Flavio Bonomi | Cisco Systems |
| Dr. Michael Campbell **(Chair)** | The Aerospace Corporation |
| Mr. Son K. Dao | HRL Laboratories, LLC |
| Mr. Don Dye | Acorn Technology Corporation |
| Petros Efstathopoulos | Symantec Corporation |
| Mr. Vikram Gupta | Qualcomm Inc. |
| Dr. John Harrell | The Aerospace Corp. |
| Mr. Arman Hovakemian | Naval Surface Warfare Center |
| Mr. Erik Hoel | Environmental Systems Research Institute |
| Mr. Yu-Chin Hsu | Novas Software, Inc. |
| Mr. Mark Jeffrey | CTO Serial Entrepreneur/ Mahalo.com |
| Dr. Ram Keralapura | Office of CTO, Narus Inc. |
| Mr. Ravi Kumar | Yahoo! Research |
| Mr. Joachim Kunkel | Synopsys, Inc. |
| Dr. Bill Luebke | Naval Surface Warfare Center |
| Dr. James R. McGraw | Lawrence Livermore National Lab |
| Dr. Scott Morehouse | Environmental Systems Research Institute |
| Dr. Ravi Iyer, | Intel Corporation |
| Mr. Sibabrata Ray | Google Inc. |
| Dr. Prabhakar Raghavan | Yahoo! Labs |
| Mr. Doug Rosen | Microsoft |
| Mr. Anthony Sarris | Unisys Corporation |
| Ms. Pat Thaler | Agilent Technologies, Inc. |
| Mr. Geoffrey O. Thompson | Nortel Networks, Inc. |
| Mr. Kees Vissers | Xilinx Research, Inc. |
| Dr. Ghaleb Abdulla | LALN |
| Dr. Ravi Iyer | Intel |
| Dr. Jim Larus | Microsoft |

The Computer Science Program is particularly sensitive to the needs of employers of our students. These employers are a diverse group, including (considering only students that graduated in 2012) defense contractors such as Raytheon Space and Airborne Systems, Rockwell Collins Inc and Boeing, communication/information heavyweights such as Ebay, Hewlett Packard, Microsoft, Google, Verizon Wireless, Environmental Systems Research Institute and SBC Communications, financial services companies including Farmers Insurance Group, and Ameriquest Mortgage Company, and numerous start-ups such as Fetch Technologies, LunarPages (Add2Net, Inc) and ACMS inc.

Table 10 illustrates how our program educational objectives meet the needs of our constituents. This mapping is shown to the entire faculty at least once a year (most recently on March 5th 2012) who are invited to discuss it and offer revisions.

Table 10: How Program Educational Objectives meet the needs of our Constituents

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Constituents  PEOs (abridged) | **Students** | **Faculty** | **Employers** | **Alumni** | **Advisory Boards** | **Community at large** |
| success in *post-graduation studies* | students clearly benefit, financially and otherwise from going on to higher education | faculty care deeply about the reputation of UCR, and our students are our most important ambassadors to other universities | many employers need employees with advanced degrees and significant networking skills | alumni benefit from a halo effect when our current students, become fellow alums and are successful in post-graduation studies and/or a chosen profession or vocation , and make significant contributions to society | note that our Advisory Boards have high overall with Alumni and Employers.  They have offered their valuable time and expertise for free, and naturally wish to see the maximum benefit extracted from it.  Our students success in post-graduation studies and/or a chosen profession or vocation , and their contributions to society, are the metrics they have suggested as a measure of the success of their input | students with post graduate degrees earn significantly more, boosting tax revenues for the community |
| success in *a chosen profession or vocation* | students clearly benefit from high career satisfaction and obtaining promotions/raises | faculty care deeply about the reputation of UCR, and our students are our most important ambassadors to industry | employers and our students mutually benefit when students have satisfying careers and are promoted/ recognized | entrepreneurial activities in a community increase the tax base |
| *contributions to society* | students benefit from even the *possibility* of public/volunteer service | faculty care deeply about the reputation of UCR, and our students are our most important ambassadors to society at large | employers benefit from the halo effect when they have employees that are engaged in public/volunteer service | the community clearly benefits from citizens anxious to engage in public/volunteer service |

1. **Process for Revision of the Program Educational Objectives[[4]](#footnote-4)**

The Computer Science degree is the product of a sequence of core and advanced courses offered by the Computer Science and Engineering Department. The department controls the process of establishing course objectives for its own courses. Table 11 outlines the general process by which we use data to improve our program. Note that *this process itself is examined once a year for meta-improvements*, and it is this meta-process that is the Process for Revision of the Program Educational Objectives. The process can be seen as an “*inner loop*”, which is conducted once a quarter, and an “*outer loop*”, which is conducted once a year.

Table 11: The Program Iterative Improvement Process. Note that one item that is considered once a year in this process are the Program Educational Objectives (*top left*)

|  |
| --- |
|  |

**The Inner Loop: Individual Course and Course Sequence Level**

The inner loop is discussed in exhaustive detail in Section 4.B, so we avoid a detailed discussion here. Suffice it to say that each quarter, for each course offering, a great deal of empirical information regarding the coverage and performance on program educational objectives are gathered, analyzed and discussed at faculty meetings.

**The Outer Loop: Curriculum and Program Level**

**Important note**: In this context, we are asking if our PEOs appropriate—do they meet constituent needs? (Are we doing the *right thing*?). We also ask ourselves are our graduates attaining the PEOs? (Are we doing *things right*?), but this is discussed in Criterion 4, not here.

At the end of each year, the following data is collected:

* Senior Exit Surveys. The survey allows the graduating seniors to rate how well the program met the objectives and outcomes. The senior Exit Surveys are distributed to the faculty and analyzed. The Undergraduate Committee then drafts an action plan for improvement.
* Board of Advisors survey. Each year, the department organizes meetings with advisory boards.
* The Undergraduate and ABET Committees are tasked with collecting and analyzing the BOA feedback on the courses content, program objectives, etc.
* Quantitative assessment of the CS 179 Senior Design projects against student outcomes.
* Alumni Surveys. These surveys are collected from the set of alumni and analyzed with the goal of determining the importance and relevance of the program objectives and outcomes, as well as their achievement.

The assessment process itself has been continually revised and improved since 2003 to incorporate more quantitative assessment elements. For example, the student Exit Surveys were originally administered in the last session of the senior design course (CS 179), but we realized that this allowed students to graduate without filling out a survey. We now administer it through the Office of Student Academic Affairs. Students must complete the exit survey when they file their applications for graduation. Graduation applications are not accepted without the survey. This assures 100% participation in the survey.

The instructor for each undergraduate course is required to keep a course file, documenting important information such as syllabus, course matrix (i.e. course objectives vs. Student Outcomes), testing/measurement information, course assessments, report, and recommendations for future improvements. While the instructor is *responsible* for this, in practice the TAs actually do most of the paperwork. They are trained for this task in the first two weeks of CS 302, Apprentice Teaching. The loop is “closed” each time a new instructor teaches the course by a mechanism we call instructor “sign-on,” a procedure whereby each new instructor reads and signs off on the recommendations made by the previous instructor (which could be the same person) for the improvements in the course curriculum.

The information in the course files is integrated and analyzed by the CS Undergraduate Instructional Committee and CS ABET committee at the end of each academic year. Additional data obtained from the industry Board of Advisors (BOA), students, and alumni, is analyzed. Based on this analysis recommendations may be made to the faculty for changes and/or improvements in the PEOs, outcomes, or any aspect of the program. If the faculty approves, the improvement actions are then propagated forward to make the recommended changes in the program.

**Changes Made to PEOs since Last ABET Accreditation**

**Note**: *Some of this information is redundant with Criterion 2.B, but included here for ease of reference*.

The last time the PEOs were changed was in 2007, below we describe this change in detail:

After the 2006/07 ABET evaluation, we received feedback that the evaluators had some suggestions about our CEN and EE PEOs (recall that our CSE department hosts CEN). In late October 2007, the two faculty members from the ABET Committee, Dr. Eamonn Keogh and Dr. Tom Payne, had a series of meetings with the interested parties, including the CE ABET committee, the EE ABET committee (at the time, Dr. Amit K. Roy-Chowdhury and Dr Roger Lake), the CS Undergrad education committee, the Chair of CS/CE, Dr. Laxmi Bhuyan, some of the most frequent employers of our students (ISCA Tech in Riverside, ESRI in Redlands), and (by phone) some of our BOA, including Dr. Mark Campbell of The Aerospace Corp.

On November the 3rd to 6th 2007, Dr. Eamonn Keogh and Dr. Tom Payne wrote new CS/CE PEOs. They based them very closely on the EE PEOS. The Electrical Engineering department was consulted and asked for feedback at this stage. On November 8th 2007, Dr. Eamonn Keogh presented the new PEOs to the Board of Advisors, from 1:00pm to 1:30pm. Each member received a take home copy, and was invited to discuss the PEOs both at the meeting, and offline by email at a later date. Dr. Keogh also discussed ABET more generally, and with Dr. Neal Young they discussed the undergraduate program in general. Almost all the faculty was in attendance. Ms. Andrea Gonzales took minutes. Figure 9 shows supporting documentation for this, the original documents are available on request.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 9: *left*) The Agenda for the BOA Meeting held on 11/8/07. *center*) and *right*) the attendance Roster at the BOA Meeting held on 11/8/07

The ABET committee then solicited feedback and approval for our new PEOs from employers of our CS students. This process was conducted by phone, email and when possible, by an onsite visit by a delegation from our department. Figure 10 documents this process with one company that has hired four Computer Science students, similar documentation for other companies is available on request.

|  |
| --- |
|  |

Figure 10: A Letter from an Employer of Several Computer Science Students, Discussing the PEO Feedback Solicitation

We also solicited comment and approval for our new PEOs from our students, this is documented elsewhere in this report, and omitted here for brevity.

On November 14th 2007, Dr. Keogh presented the new PEOs with notes from the Board of Advisors to the entire faculty, 40 minutes were spent discussing the PEOs and they were adopted by a majority vote. The minutes of this meeting are detailed in Figure 11.

|  |
| --- |
| Faculty Meeting Minutes from November 14, 2007  The meeting commenced at 12:00 p. m. by Chair Laxmi Bhuyan. The following faculty were present: Laxmi Bhuyan, Rajiv Gupta, Tao Jiang, Mart Molle, Walid Najjar, Chinya Ravishankar, Vassilis Tsotras, Michalis Faloutsos, Eamonn Keogh, Srikanth Krishnamurthy, Stefano Lonardi, Thomas Payne, Neal Young, Harry Hsieh, and Christian Shelton.  (A) Announcements were made by Laxmi regarding:  (1) The faculty meeting will be held every two weeks from 12-1 p. m.  (2) No corrections were suggested for the October 24th meeting minutes.  (3) Intel will be sending from $10,000-$20,000 to support the distinguished lecture series and their logo will appear on the series posters.  (4) Mike Carrey has been recommended by Reza for an Eminent Scholar position at UCR- Discussion took place as to where he is interested in applying for a position and what we can do to make an attractive offer to him. Mike will be back to Riverside December 7th for a visit.  (B) **Eamonn presented the proposed ABET program educational objectives and their need to be measurable and explicit. Discussion ensued as to whether or not they should be published in the general catalogue. A vote was taken and the faculty voted for the Program Educational Objectives as presented.**  (C) Debate was held about the Advisory Board and what their role is in relation to the Department; their interest is in graduate and undergraduate programs. Laxmi suggested that perhaps there should not be an Advisory Board meeting every year. He stated that he had presented the agenda to faculty for feedback and received none. Perhaps the agenda should be different next year and change it to an Industry Day. Or perhaps have a separate Industry Day and keep the Board of Advisors for ABET purposes. It was suggested that if the department held an industry day there could be more research and poster presentations which would feature graduate student research. The population of the board of advisors was discussed and that most were high level managers. It was suggested that perhaps it would be better to target industry affiliates who are mid level managers and would be the managers of the people who would actually hire our students. The department could form a committee to update the list (which hasn’t been reviewed and updated since 2001) by deleting old inactive names and inviting new people. Due to time constraints it was suggested that this discussion be continued at another meeting.  (D) Christian Shelton’s merit was discussed and ballots were distributed for voting.  (E) Vassilis said it would be good to have 4:00 p. m. meetings on Fridays for students to meet with the faculty.  The meeting was adjourned at 1:40 p.m. |

Figure 11: Computer Science and Engineering Faculty Meeting Minutes from Nov 14, 2007

In summary, we created new PEOs after soliciting input and approval from all our constituents, our Board of Advisors, employers of our students, faculty, and the students themselves. Furthermore we have *carefully documented* this process at every step.

**Summary for this Section**

As illustrated in Table 11, we have a detailed and rigorous process for review and possible revision of the Program Educational Objectives. This process takes place once a year, and input is obtained from *all* constituents. At least two hours a year (at the faculty retreat in September) are reserved for *the entire* faculty to discuss the PEOs face-to-face.

# CRITERION 3. STUDENT OUTCOMES

1. **Student Outcomes**

After consulting with all our constituents, the faculty decided to adopt the following as our Student Outcomes in 2009. We revisit this issue at least once a year at the faculty retreat (September), however thus far we have found no reason to change or augment the outcomes.

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

The reader will readily recognize the origin of these outcomes. Outcomes I to IX are the CAC (a) to (i) student outcomes. Student outcomes X and XI are the additional *Student Outcomes for Program Criteria for Computer Science and Similarly Named Computing Programs*, (j) and (k). Finally student outcomes XII and XIII are EAC student outcomes (b) and (j) respectively.

The reader may wonder why we adopted student outcomes XII and XIII. Part of the answer is pragmatic. For all the courses in the CS program that are also used by either the CEN program or the EE program (just EE140 in the latter case) we must measure these outcomes for those programs. Thus we simply have measurements of coverage and performance on these outcomes “for free”. More importantly however, the faculty and our Board of Advisors strongly feel that an *ability to analyze and interpret data* and *knowledge of contemporary issues* are fundamental skills for *any* scientist, including computer scientists.

It is very important to note that unlike the original EAC ABET outcomes, our student outcomes *do* have some redundancy. In particular, student outcome X can be seen as a combination of student outcomes I, II and III, and student outcomes XI can be seen as a combination of student outcomes III and IX. We recognize these redundancies, and carefully guard against overcounting evidence when analyzing data (cf. Criterion 4). However we have done this for two reasons:

* Pragmatically, this allows us to collect data from the many courses that serve both CEN and CS in *a single unified system that maximizes simplicity and compliance*. Only *after* the data is collected do we need “factor it” into evidence for CS or CEN.
* In some cases, we have found it useful to have two slightly different “views” of the same data during analysis and discussion of performance and coverage of student outcomes.

For reference, we review our student outcomes and their relationship to the ABET CAC and EAC outcomes in Table 15.

**Table 15: The mapping between the UCR CS Student Outcomes, and the CAC Student Outcomes with Program Criteria for Computer Science, and the EAC Student Outcomes**

|  |  |  |
| --- | --- | --- |
| **CS Student Outcomes** | **CAC Student Outcomes with Program Criteria for Computer Science** | **EAC Student Outcomes** |
| I (from CAC) | (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline | (a) an ability to apply knowledge of mathematics, science, and engineering |
| II (from CAC) | (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution | (e) an ability to identify, formulate, and solve engineering problems |
| III (from CAC) | (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs | (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability |
| IV (from CAC) | (d) An ability to function effectively on teams to accomplish a common goal | (d) an ability to function on multidisciplinary teams |
| V (from CAC) | (e) An understanding of professional, ethical, legal, security and social issues and responsibilities | (f) an understanding of professional and ethical responsibility |
| VI (from CAC) | (f) An ability to communicate effectively with a range of audiences | (g) an ability to communicate effectively |
| VII (from CAC) | (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society | (h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context |
| VIII (from CAC) | (h) Recognition of the need for and an ability to engage in continuing professional development | (i) a recognition of the need for, and an ability to engage in life-long learning |
| IX (from CAC) | (i) An ability to use current techniques, skills, and tools necessary for computing practice. | (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. |
| X (from CAC  Student Outcomes for Program Criteria for Computer Science ) | (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. |  |
| XI (from CAC  Student Outcomes for Program Criteria for Computer Science) | (k) An ability to apply design and development principles in the construction of software systems of varying complexity. |  |
| XII (from EAC) |  | (b) an ability to design and conduct experiments, as well as to analyze and interpret data. |
| XIII (from EAC) |  | (j) knowledge of contemporary issues. |

We note that our department has a culture of openness to new ideas, and constantly researches and discusses our student outcomes. We do this not only using the formal process we design for this task (cf. Table 18 in Criterion 5), but also by including ad-hoc information from news reports, white papers, proceedings of journals and conferences on engineering education etc. As a representative example, at the May 9th 2012 faculty meeting we discussed at length the findings of a 2009 national survey by the Association of American Colleges and Universities, “*Raising The Bar: Employers Views On College Learning In The Wake Of The Economic Downturn*”[[5]](#footnote-5). This report provided us with useful information indicating our student outcomes are relevant to employers. Quoting from the survey: ―*The areas in which employers feel that colleges most need to increase their focus include*

1) *written and oral communication*

2) *critical thinking and analytical reasoning*

3) *the application of knowledge and skills in real-world settings*

4) *complex problem-solving and analysis*

5) *ethical decision-making*

6) *teamwork skills*

7) *innovation and creativity*

8) *concepts and developments in science and technology*

Table 13 maps these employer responses to our student outcomes, an exercise we completed at the faculty meeting on May 9th 2012 to spur further discussion of our student outcomes.

Table 13: The Relationship of Student Outcomes to AAC&U Employer Survey

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AACU survey  Student Outcomes | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| I. An ability to apply knowledge of mathematics, science, and engineering |  |  |  |  |  |  |  |  |
| II. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution |  |  |  |  |  |  |  |  |
| III. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs |  |  |  |  |  |  |  |  |
| IV. An ability to function effectively on teams to accomplish a common goal |  |  |  |  |  |  |  |  |
| V. An understanding of professional, ethical, legal, security and social issues and responsibilities |  |  |  |  |  |  |  |  |
| VI. An ability to communicate effectively with a range of audiences |  |  |  |  |  |  |  |  |
| VII. An ability to analyze the local and global impact of computing on individuals, organizations, and society |  |  |  |  |  |  |  |  |
| VIII. Recognition of the need for and an ability to engage in continuing professional development |  |  |  |  |  |  |  |  |
| IX. An ability to use current techniques, skills, and tools necessary for computing practice. |  |  |  |  |  |  |  |  |
| X. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. |  |  |  |  |  |  |  |  |
| XI. An ability to apply design and development principles in the construction of software systems of varying complexity. |  |  |  |  |  |  |  |  |
| XII. An ability to design and conduct experiments, as well as to analyze and interpret data. |  |  |  |  |  |  |  |  |
| XIII. Knowledge of contemporary issues. |  |  |  |  |  |  |  |  |

One outcome of this discussion was a consensus that we need to redouble our efforts in *measuring* the student’s teamwork skills. Currently the assessment of this skill is limited to peer-assessments by the team, observations by the instructor, and how well the team performs on the given task (especially in CS179, our capstone project). It was decided that a committee would be formed to attempt to find additional metrics for teamwork skill that could be incorporated into various instruments.

1. **Relationship of Student Outcomes to Program Educational Objectives**

Table 14 shows the relationship of Student Outcomes to Program Educational Objectives. This mapping was created in late October 2008 by the CS ABET Committee, and shown to the full faculty for comments and discussion on November 12th 2008.

Table 14: The Relationship of Student Outcomes to Program Educational Objectives

|  |  |  |  |
| --- | --- | --- | --- |
| PEOs (abridged)  Student  Outcomes | **Success in post-graduation studies *as evidenced by*:** | **Success in a chosen profession or vocation *as evidenced by*:** | **Contributions to society *as evidenced by*:** |
| I. An ability to apply knowledge of mathematics, science, and engineering | advanced degrees earned | career satisfaction/ promotions & raises |  |
| II. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution |  | entrepreneurial activities /  consulting activities |  |
| III. An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs |  | career satisfaction/ promotions & raises | public service |
| IV. An ability to function effectively on teams to accomplish a common goal | professional responsibilities | entrepreneurial activities | leadership roles |
| V. An understanding of professional, ethical, legal, security and social issues and responsibilities |  |  | volunteer service/ mentoring / outreach activities/ public service |
| VI. An ability to communicate effectively with a range of audiences | professional visibility | promotions & raises / professional visibility / consulting activities | mentoring / outreach activities |
| VII. An ability to analyze the local and global impact of computing on individuals, organizations, and society | professional responsibilities |  |  |
| VIII. Recognition of the need for and an ability to engage in continuing professional development | satisfaction with the decision to further their education/ professional visibility | professional visibility |  |
| IX. An ability to use current techniques, skills, and tools necessary for computing practice. | professional visibility | entrepreneurial activities /  consulting activities |  |
| X. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. |  | entrepreneurial activities /  consulting activities |  |
| XI. An ability to apply design and development principles in the construction of software systems of varying complexity. |  | career satisfaction/ promotions & raises |  |
| XII. An ability to design and conduct experiments, as well as to analyze and interpret data. |  | career satisfaction/ promotions & raises |  |
| XIII. Knowledge of contemporary issues. |  | entrepreneurial activities /  consulting activities | mentoring / outreach activities/ public service |

# CRITERION 4. CONTINUOUS IMPROVEMENT

We begin with Table 15 which shows a visual overview of the process that we use to assess and evaluate the extent to which the Program Educational Objectives and Student Outcomes are being attained, and make improvements. While both processes are holistic, the “*inner loop*” focuses on the Student Outcomes and the “*outer loop*” focuses on the Program Educational Objectives.

Table 15: The Continuous Improvement Process. Note that this table is similar to Table 11, however there we consider the process for the review and possible revision of the correctness of the Program Educational Objectives, but here we are considering how we assess and evaluate the attainment of the *Program Educational Objectives* and the *Student Outcomes*.

|  |
| --- |
|  |

**The Inner Loop: Individual Course and Course Sequence Level**

At the end of each quarter, the following data are collected:

* Grades in homework assignments, lab reports, short tests and examinations. Review of the student performance (grade received) for feedback on whether the course objectives and Student Outcomes are met. All the material is archived in a standardized format (see “*the ABET Binder*” in Section 4.B below), to simplify inspection (by ABET evaluators) and analyses (by faculty).
* Student Evaluation of Teaching. Evaluations administered near the end of each quarter allow students to provide the instructor with anonymous feedback on the effectiveness of the course. The questions in the evaluation forms include questions relevant to the stated program objectives like “*Have you learned something you consider valuable*?”
* End-of-course student assessments/surveys. Course surveys are distributed at the end of each course. The course survey is based on the course objectives, and Student Outcomes I 🡪 XIII from the course objective matrix. Students are asked how well the course objectives, and Student Outcomes were achieved.

**The Outer Loop: Curriculum and Program Level**

At the end of each year, the following data are collected:

* **Senior Exit Surveys**. The survey allows the graduating seniors to rank how well the program met the objectives and outcomes. The senior Exit Surveys are distributed to the faculty and analyzed. The Undergraduate Committee then drafts an action plan for improvement.
* **Board of Advisors Surveys**. Each year, the departments organize meetings with industry advisory boards. The Undergraduate and ABET Committees are tasked with collecting and analyzing the BOA feedback on the courses content, Program Educational Objectives, etc.
* **Senior Design Projects** quantitative assessment of the CS 179 Senior Design projects (discussed in great detail in the Curriculum section of this document).
* **Alumni Surveys**. These surveys are collected from of alumni and analyzed with the goal to determine the importance and relevance of the Program Education Objectives and Student Outcomes, in addition to their achievement. The last two surveys were conducted in Spring 2008 and Spring 2012. The ABET Chair sends out the surveys by email to the alumni based on their latest contact information that is available. This cycle we began to explore using social media to further improve our response rate. The survey consists of questions that directly measure different aspects of the PEOs (e.g., whether the alumni has completed a graduate degree, whether the alumni has completed a professional certification), as well as more indirect questions such as how well they were prepared for their career choice.
* **Faculty Input**: Inputs received from the Alumni and Board of Advisors are analyzed by the undergraduate committee, in the context of the other sources of information listed above. A summary is then presented to the faculty with a list of modifications that should be implemented for better achievement of the PEOs.

Having seen an overview of the entire process, in the next two sections we will consider the PEOs (*outer loop*) and course objectives (*inner loop*) in great detail.

1. **Program Educational Objectives**

We use several mechanisms to measure how well the program is achieving the PEOs. These include employer surveys, employer visits and interviews, alumni surveys, and industrial advisory committee meetings. Below we consider each in turn.

*4.A.1 Consulting with our Board of Advisors on PEOs*

Our department maintains a close relationship with our board of advisors. We meet with them formally once a year (typically early in the winter quarter) for a full day, and we also correspond with them on faculty visits and through informal email and phone surveys.

The last meeting with our board of advisors took place on April 2nd 2012. While seventy-five minutes was allotted to undergraduate education, the discussion of undergraduate issues spilled into the heavily attended working lunch and dinner sessions. The unedited notes taken at the meeting were further discussed at a faculty meeting that took place on April 4th 2012.

|  |  |
| --- | --- |
| (*unedited*) Minutes of the discussion during the Board of Advisors,  Dept of Computer Science and Eng, April 2nd 2012. Scribe:  Prof. Stefano Lonardi, Vice Chair and Professor.  Discussion started from the Board of Advisors raising  the issue of the need for undergrad students of having  Calculus in their curriculum for CS and CE degree.  Someone in BOA mentioned that perhaps more discrete math (concept of proof, induction, correctness) would be more appropriate than Calculus. Others in BOA mentioned that Statistics, Logic, Communication Theory (Information Theory) or even Economics would be more appropriate: perhaps broadening the range of course undergraduate could take would be beneficial for some students (e.g. replacing Calculus+Physics with other options).  A discussion about the "problem" with the quality of teaching for math and physics at UCR followed: often for math these course are taught by visiting professor, that are not very interested in offering a high quality learning experience for students (Prof. Neal Young and Frank Vahid).  Some advisors mentioned that we should incorporate in the math background skills that allow student to "turn data into knowledge" -- i.e., probability and statistics.  These days many disciplines are, or are becoming computational. The next topic of discussion was about retention. Someone in the BOA mentioned that broadening the experience in the math series by offering alternatives to Physics + Calculus could help increasing retention. Someone said that "perhaps not all CS students are interested in learning about nuts and bolts".  The last topic was about our choice of the programming language used to teach freshmen introduction to programming, which is C++. Some BOA member felt that C++ is not the best language to learn as the first programming language: too unforgiving and frustrating due to its syntax. Prof. Neal Young explained the rationale behind C++, that we have discussed alternatives, but some upper division classes (OS, networks, etc) need C++.  Faculty that participated in the discussion: Prof. Neal Young, Prof. Frank Vahid, Prof. Marek Chrobak, Prof. Tao Jiang, Prof. Laxmi Bhuyan, Prof. Tamar Shinar, Prof. Michalis Faloustos, Prof. Stefano Lonardi, Prof Eamonn Keogh. |  |

Figure 12: *left*) minutes from the Advisory Board meeting that took place on April 2nd 2012 shows that well over an hour (*right*) was devoted to undergrad education issues.

BOA in attendance for the full day were: Ghaleb Abdulla -LALN, Mike Campbell (chair) - Aerospace Corp, Don Dye - Acorn, Petros Efstathopoulos - Symantec, Vikram Gupta - Qualcomm, John Harrell - Aerospace Corp, Ravi Iyer - Intel, Jim Larus - Microsoft and Kees Vissers - Xilinx.

In Figure 13 we show the notes taken at this meeting by the CS BOA chair, Dr. Mike Campbell of the Aerospace Corp. These notes with Dr. Lonardi’s notes (Figure 12) were discussed at length at a faculty meeting on April 18th 2012.

|  |
| --- |
| **Chair's opening remarks:**  Prof. Bhuyan laid out the Department's current structure, priorities, and goals. This presentation and all the presentations throughout the day showed effective focus on the Department's three primary goals: 1. Continued enhancement of the Department's national ranking; 2. Maintaining high quality faculty, graduate students, & undergrads; and 3. Enhancing Faculty visibility and recognition. In spite of the current budget constraints, the Department is maintaining exceptionally high productivity, especially in terms of PhDs per Faculty per year. The Department seems to be weathering the current budget issues well. This is a credit to the leadership and the willingness of the Faculty and Staff to do whatever it takes.  **Graduate Program:**  Prof. Jiang's presentation led to a very good discussion of the current balance between the MS and PhD programs. There are many issues, and the discussion provided a very strong rationale for the current structure. In particular, the Chair and the Faculty made a very strong case that the current MS/PhD balance is appropriate in support of the Department's three most important goals, above. (This is independent of the current State budget issues.) It appears that increasing the number of MS degree students would only be advisable as a very long-term goal and may require structural changes in the way funding is allocated to Departments for teaching resources.  **Undergraduate Program Discussion Topics:**  **Undergrad retention:**  The statistic "30-40%" may be misleadingly low. Need to present that statistic with enough definition, context, and benchmarking vs. other institutions. That said, undergrad retention is still an important metric of the Department's success.  **Undergrad core curriculum:**  Advisory Board members questioned the six-course Math requirement. The number of courses may be appropriate; however, on the surface it appears to be heavily weighted towards continuous mathematics and Calculus in particular. The rationale for this is that the students need to learn mathematical rigor and how to construct a formal proof.  There was a consensus by the Board that the requirement for learning rigor and how to write proofs is an important and relevant goal. Our discussions revealed that the overall course content is actually more balanced than it first appeared and does include coverage of discrete math and statistics, especially in the latter required courses and in the recommended electives. It was widely noted that this breadth of mathematical course work is a very positive thing. However, there was a near consensus that there are other lower division Math courses besides Calculus that could fulfill the goal of teaching theorem proving. It may be beneficial for this required core Math sequence to place even greater emphasis on discrete math, statistics, probability, and quantitative analysis. This might require decreased emphasis on Calculus, and it may be beneficial to give students more options for how to meet the course requirements aimed at teaching theorem proving.  There were also questions re. the one-year Physics requirement. It is understood that this course requirement reflects the relatively heavy emphasis on hardware in this Department. The Board supports this emphasis. In fact, the basic science requirement can serve two purposes: 1) understanding physical computer hardware, and 2) strengthening the students' ability to apply computational technology to scientific applications. However, several Board members questioned whether three quarters of basic Physics is the best complementary coursework to support an understanding of computer hardware. There was a near-consensus that this Physics sequence requirement should be broadened to allow for other areas of science, including both physical sciences (biology, chemistry, environmental) as well as behavioral & social sciences (psychology, economics). Broadening the science requirement in this way would also serve many of the Department's other goals for their undergrads. It was noted that there are aspects re. the instruction of the current Physics sequence by outside faculty that influence the implementation of this requirement. If the suggestion to broaden the science requirement is implemented, it is understood that these concerns with instruction by outside faculty will have to be addressed again & again for each new area, which will require additional University resources and significant collaboration with other departments.  **ABET:**  Prof. Keogh's presentation on the ABET process gave a good introduction for those Board members that were new to the topic, and provided a valuable update for those of us that have been on the Board since before the last accreditation. The thoughtful planning that has gone into this by Prof. Keogh and the entire Faculty, Staff, and Administration really shows. I'm sure I speak for all of the Board when I say that we understand the importance of this process, and we are all willing to support your ABET work in any way we can.  **James Larus' Distinguished Speaker Lecture:**  Dr. Larus' presentation was very well done, and it was clearly very well received by the Advisory Board as well as the general audience of faculty and students. There was some discussion of whether it is best to schedule the DSL during the Advisory Board meeting. The general consensus seemed to be that overall this benefits the goals of the Advisory Board meeting.  **Intro. of New Faculty:**  Prof. Madhyastha gave a very good presentation on improving understanding of how applications behave in a thin client environment. This research demonstrated very good breadth by addressing the full spectrum from low level behaviors & metrics to the user experience.  Prof. Shinar enthusiastically described her very interesting and timely work on physics-based modeling for both computational science and graphics applications. Also described her work with young people to encourage females (and other under-represented groups) to enter STEM and be successful. This is recognized as a very important long-term goal by the Board members. She is to be commended for dedicating her personal time and energy to it.  Prof. Hristidis' presentation re. his team's work in Databases / Data Mining, one of the Department's traditional strengths, demonstrated very good balance between theory and application. |

Figure 13: The CS BOA Chair, Dr. Mike Campbell, wrote these (unedited) minutes at the Advisory Board meeting that took place on April 2nd 2012

*4.A.2 Consulting with Employers on the appropriateness and achievement of PEOs*

We begin by noting in passing that our BOA (discussed in previous section) and our student employers have a large overlap, although we deliberately make sure it is not a perfect overlap.

One method we use to gather information on *both*...

1) ..how well are our graduates meeting the Program Educational Objectives...

2) ..how well the current Program Educational Objectives align with employers needs...

...exploits the fact that our faculty frequently visit our graduates’ employers.

Some representative examples from the last two years include: Dr. Shelton, NSWC Corona (Naval Surface Warfare Center, Corona), Dr. Faloutsos, The Bourns Company (not to be confused with Bourns College of Engineering)., Dr. Brisk Intel Corp, Western Digital., Dr Payne, Jethead Development, Luminex, Inc, Aerospace Corp., Dr. Najjar, ISCA Technologies, Jacquard Computing., Dr Keogh, ISCA Technologies, Microsoft, Google, ESRI., Dr Vahid, Qualcomm., Microsoft, Google, Altera Corp Western Digital etc.

We have created a one-page (double sided) flyer that we use to prompt conversations with our graduates’ employers. Figure 14 shows its contents.

|  |
| --- |
|  |

Figure 14: The PEO feedback solicitation flyer (“The Orange Sheet”)

This flyer is printed on bright orange paper (to make it visually memorable) and placed in every faculty member’s physical mailbox twice a quarter. Faculty are encouraged to take the flyer with them on visits to employers and use it as a conversation starter on PEOs. While this flyer considers “*How well the current Program Educational Objectives align with employers needs.”* it also invariably spurs some conversation on “*How well are our students meeting the Program Educational Objectives*.”. The information gleaned by this direct contact, either face to face with the visiting faculty, or sent offline to the chair, is discussed at the next faculty meeting, and all such feedback is summarized by the ABET chair (in this cycle, Dr. Keogh) at the annual faculty retreat.

*4.A.3 Alumni Survey*

We conduct an alumni survey every two years, the last survey was conducted in Spring 2012. The number of responses was 49.

In Table 16 we show the results of the 2012 survey, compared to the 2008 survey, which is the last survey that was seen by ABET evaluators.

Table 16: The CS Alumni Survey, annotated by relevance to our Program Educational Objectives and to the Student Outcomes (SO), contrasting the results of the 2008 survey and the 2012 survey.

|  |  |  |
| --- | --- | --- |
| Key | Questions answered by students that have gone to grad school | Questions answered by students that have *not* gone to grad school |

|  |
| --- |
| **1. What year did your earn your bachelor's degree in Computer Science?**  (Bookkeeping question only)  2007(**4**), 2006(**0**), 2005(**8**), 2004(**1**), 2003(**1**), 2002(**3**):  2012(**2**), 2011(**3**), 2010(**4**), 2009(**2**), 2008(**6**), 2007 or before(**32**):  Note that the response rate seems to have increased, due to more aggressive tracking of alumni |
| **2. Have you pursued or completed any degrees beyond your bachelor's degree from UCR?**  **Yes, No**  (Bookkeeping question only, to bifurcate the following questions into those that have had graduate education and those that have not)  56.3% said **yes**:(2008)  53% said **yes**:(2012)  It is very satisfying that half of respondents have pursued or completed advanced degrees. |
| **3. If you have completed another degree, please indicate all degrees completed.**  **M.S., Ph.D., MBA, J.D., M.D. Other (please specify)**  (This question maps onto SO VIII directly, and to directly onto PEO “*satisfaction with the decision to further their education*” and “*advanced degrees earned*”)  MS 57%, PHd 43%. (2008)  MS 72%, PhD 28% (2012) |
| **4. If you are pursuing another degree, please indicate the degree you are pursuing.**  **M.S., Ph.D., MBA, J.D., M.D.**  **Other (please specify)**  (This question maps onto SO VIII directly, and to directly onto PEO “*satisfaction with the decision to further their education*” and “*advanced degrees earned*”) Since these choices are not mutually exclusive, they may sum to <100%  MS 66.7%, PhD 50.0% (2008)  MS 27%, PhD 82% (2012)  While we recognize that possibility of selection bias, we are delighted to find that fully 4/5ths of our students are pursuing PhDs |
| **5. Have you published articles and/or made presentations at conferences in your field?**  **Yes, No**  (This question maps onto SO I, XII, VI, and III directly, and to directly onto PEO “*success in post-graduation studies as evidenced by professional visibility*”)  44.4% **Yes** (2008)  61.53% Yes (2012)  This metric seems to have improved significantly. |
| **6. Have you been named on any patents or patent applications?**  **Yes, No**  (This question maps onto SO III, VI, XIIII, I and IX directly, and to directly onto PEO “*success in post-graduation studies as evidenced by professional visibility-patents*”)  O% **Yes**. (2008)  19.23% Yes (2012)  This metric seems to have improved significantly. |
| **7. Have you received any awards for professional achievement? Yes, No, If yes, please describe**  (This question maps onto SO I and II directly, and to directly onto PEO “*success in post-graduation studies as evidenced by professional visibility-awards*”)  22% **Yes** (2008)  23.3% Yes (2012) |
| **8. Have you engaged in any international research or collaborations (e.g., presented at international conferences, worked with international collaborators)? Yes, No, If yes, brief description of international activities**  (This question maps onto SO IV, VI and VIII directly, and to directly onto PEO “*success in post-graduation studies as evidenced by professional visibility*”)  22% **Yes** (2008)  46.15% Yes (2012)  This metric seems to have improved significantly. |
| **9. Have you been a program committee member or organizing committee member of a conference?**  **Yes, No, If yes, how many times?**  (This question maps onto SO V, VI, VIII, VII, and XIII directly, and to directly onto PEO “*success in post-graduation studies as evidenced by professional visibility-reviewing and editorial work for professional journals*”)  22% **Yes.** (2008)  11.53% Yes (2012) |
| **10. Have you been a reviewer for any journals? Yes, No, If yes, approximately how many times?**  (This question maps onto SO V, VI, VIII, VII, and XIII directly, and to directly onto PEO “*success in post-graduation studies as evidenced by professional visibility-reviewing and editorial work for professional journals*”)  11% **Yes** (2008)  44.44% Yes (2012) |
| **11. Have you engaged in other professional service such as scientific or technical review panels, serving as an expert witness, or consulting?**  **Yes, No**  (This question maps onto SO V, VI, VIII, VII, and XIII directly, and to directly onto PEO “*Public service, leadership roles*” and “*consulting activities*”)  22% **Yes** (2008)  20% Yes (2012) |
| **12. At this point of your career and education, what is the level of your satisfaction with your career choice and success in each of the following?**  **2008 Very satisfied - 5 4 3 2 1- Not satisfied**  **The field you work in:** 66% scored a ‘5’, and 33% scored a ‘4’  **The academic institution/lab you work in:** 75% scored a ‘5’, and 25% scored a ‘4’  Recognition of your work: 22% scored a ‘5’, 44% scored a ‘4’ and 33% scored a ‘3’  **2012 Very satisfied - 5 4 3 2 1- Not satisfied**  **The field you work in:** 60% scored a ‘5’, and 36% scored a ‘4’, 4% scored a ‘3’  **The academic institution/lab you work in:** 45.83% scored a ‘5’, and 33.33% scored a ‘4’, 16.67% scored a ‘3’  Recognition of your work: 24% scored a ‘5’, 48% scored a ‘4’ and 20% scored a ‘3’  (This question maps onto SO VIII directly, and to directly onto PEO “*Satisfaction with the decision to further their education*” and “*professional visibility*”) |
| **13. While pursuing an advanced degree, have you also been working professionally?**  **I have only been a student -- skip to Professional and Community questions**  **I have worked professionally**  (Bookkeeping question only, to bifurcate the following questions into those that have had graduate education and those that have not)  62.5% of the students responded “*I have worked professionally*” (2008)  83.33% of the students responded “*I have worked professionally*” (2012) |
| **14. From the options below, please choose the one that best describes your current work.**  **CS support, CS development, CS management, CS research, Technical sales/marketing, Other (please specify)**  (This question maps onto SO VIII directly, and to directly onto PEO “*Satisfaction with the decision to further their education*” and “*leadership roles*”, and “*public service*”)  (2008)  support 0%, development 53%, management 15%, research 15%, Technical sales/marketing 7% Other (please specify) 15%.  (2012)  support 12%, development 56%, management 4%, research 4%, Technical sales/marketing 0% Other (please specify) 24%. |
| 15. At this point of your career, what is the level of your satisfaction with your career choice and success in each of the following?  The field you work in, The organization you work in, Your salary, Recognition of your work  (This question maps onto SO I *indirectly*, and to directly onto PEO “*Satisfaction with the decision to further their education*” and “*leadership roles*”, and “*professional visibility*”)  (2008)  The field you work in: 54% scored a ‘5’, and 23% scored a ‘5,’ and 15% scored a ‘3’, and 7% a ‘2’  Your organization: 54% scored a ‘5’, and 33%, scored a ‘4’ and 7% scored a ‘3’ and ‘1’  Your salary: 23% scored a ‘5’, and 30%, scored a ‘4’ and 30% scored a ‘3’, and 15% scored a ‘1’  Recognition of your work: 30% scored a ‘5’, & 30%, scored a ‘4’ & 23% scored a ‘3’, & 7% scored a ‘2’, & 7% scored a ‘1’’  (2012)  The field you work in: 52% scored a ‘5’, and 24% scored a ‘4,’ and 16% scored a ‘3’, and 4% a ‘2’  Your organization: 48% scored a ‘5’, and 20%, scored a ‘4’ and 28% scored a ‘3’  Your salary: 44% scored a ‘5’, and 16%, scored a ‘4’ and ?% scored a ‘3’, and ?% scored a ‘1’  Recognition of your work: 36% scored a ‘5’, and 28%, scored a ‘4’ and 20% scored a ‘3’  While these numbers have mostly held steady, we note the economy now (Spring 2012) is worse than four years ago. Moreover, the satisfaction with salary *does* seem to have increased significantly. This also seems to be reflected with the answers to the next question. |
| **16. Have you had promotions and/or raises since beginning your professional career?**  **Yes, No, If yes, how many times?**  (This question maps onto SO I, XII, and III *indirectly*, and to directly onto PEO “*Satisfaction with the decision to further their education*” and “*promotions raises*”)  67% **Yes** (2008)  84% **Yes** (2012) |
| **17. Have you published articles or made presentations in your organization or in your profession?**  **Yes, No**  (This question maps onto SO VI, VIII and XIII directly, and to directly onto PEO “*Professional visibility-presentations-publications*”)  41% **Yes** (2008)  44% **Yes** (2012) |
| **18. Have you engaged in international activities such as participation in international conferences, collaborative research, or employment abroad?**  **Yes, No**  (This question maps onto SO VIII, VI and IV directly, and to directly onto PEO “*Professional visibility*”)  16% **Yes** (2008)  20% **Yes** (2012) |
| **19. Have you made inventions and/or been listed on patents or patent applications?**  **Yes, No**  (This question maps onto SO III, VI, XIII, I and IX directly, and to directly onto PEO “*success in chosen profession or vocation as evidenced by professional visibility-patents*”)  0% **Yes** (2008)  16.13% **Yes** (2012) |
| **20. Have you been nominated for any professional or academic awards?**  **Yes, No, If yes, please describe. How many? Did you win?**  (This question maps onto SO I and II directly, and to directly onto PEO “*success in chosen profession or vocation as evidenced by professional visibility-awards*”)  25% **Yes** (2008)  13% **Yes** (2012) |
| **21. Have you mentored others, either inside or outside your organization?**  **Yes, No**  (This question maps onto SO VI, IV, V, and XIII directly, and to directly onto PEO “*mentoring/outreach services*” and “*Volunteer services*”)  75% **Yes** (2008)  64.5% **Yes** (2012) |
| **22. Have you led groups or teams on projects or new initiatives?**  **Yes, No**  (This question maps onto SO VI, VII and IV directly, and to directly onto PEO *“entrepreneurial activities*”, “*professional visibility*” and “*leadership roles*”)  50% **Yes** (2008)  64.5% **Yes** (2012) |
| **23. Have you engaged in any start-up businesses or been involved in any new ventures in your organization?**  **Yes, No, If yes, please describe**  (This question maps onto SO II and VII directly, and to directly onto PEO “*entrepreneurial activities*”, “*leadership roles*”)  33% **Yes** (2008)  40% **Yes** (2012) |
| **24. In the past year, have you attended any workshops, conferences, symposia, etc., related to your profession?**  **Yes, No**  (This question maps onto SO VIII directly, and to directly onto PEO “*professional responsibilities*”)  50% **Yes** (2008)  48.4% **Yes** (2012) |
| **25. In the past year, have you taken any classes/courses related to your profession?**  **Yes, No, If yes, roughly how many?**  (This question maps onto SO VIII directly, and to directly onto PEO “*professional responsibilities*”)  50% **Yes** (2008)  45% **Yes** (2012) |
| **26. In the past year, have you read any books related to your profession?**  **Yes, No, If yes, roughly how many?**  (This question maps onto SO VIII directly, and to directly onto PEO “*professional responsibilities*”)  91% **Yes** (2008)  81% **Yes** (2012) |
| **27. Do you subscribe to any periodicals related to your profession?**  **Yes, No, If yes, roughly how many?**  (This question maps onto SO VIII directly, and to directly onto PEO “*professional responsibilities*” and indirectly to PEO “*professional society membership*”)  75% **Yes** (2008)  42% **Yes** (2012) |
| **28. Are you a member of any professional societies?**  **IEEE, ACM, SAE, Other (please specify)**  (This question maps onto SO VIII directly, and to directly onto PEO “*professional responsibilities*” and indirectly to PEO “*professional society membership*”)  (2008) 100% IEEE, 40% ACM, 0% SAE, and 10% ASQ (American Society of Quality) and 10% NSBE  (2012) 33.33% IEEE, 50% ACM, 3.3% SAE, and 13.33% other |
| **29. Have you obtained Professional Engineer certification?**  **Yes, No, If no, are you pursuing PE certification, or do you plan to?**  (This question maps onto SO VIII directly, and to directly onto PEO “*professional responsibilities*” and “*satisfaction with the decision to further their career*”)  0% **Yes** (2008)  0% **Yes** (2012) |
| **30. Have you been involved in any of the following activities?**  **Public service, Community leadership roles, Volunteer activities, Mentorship and outreach activities**  (This question maps onto ABET (f)(i) and (g) directly, and to directly onto PEO “*professional responsibilities*” and “*Volunteer service/outreach activities/public service*”)  (2008) **Public service:** 64% **Yes**  **Community leadership roles:** 57% **Yes**  **Volunteer activities:** 57% **Yes**  **Mentorship and outreach activities:** 61% **Yes**  (2012) **Public service:** 43% **Yes**  **Community leadership roles:** 33% **Yes**  **Volunteer activities:** 61% **Yes**  **Mentorship and outreach activities:** 40% **Yes** |
| **31. How would you rate the importance of the following items on your career path?**  (2008) **Very satisfied- 5 4 3 2 1- Not satisfied**  **Basic math and science** 71% (**5**), 21% (**4**), 0.0% (**3**), 0% (**2**), 7% (**1**)  **… beyond math and science** 28% (**5**), 14% (**4**), 35% (**3**), 14% (**2**), 7% (**1**)  **Core curriculum in your major** 35% (**5**), 57% (**4**), 7.0% (**3**), 0.0% (**2**), 0.0% (**1**)  **Technical electives** 64% (**5**), 28% (**4**), 0.0% (**3**), 0.0% (**2**), 7% **(1**)  **Senior Design Project** 57% (**5**), 21% (**4**), 14% (**3**), 0.0% (**2**), 7% (**1**)  (2012) **Basic math and science** 74% (**5**), 27% (**4**), 4.3% (**3**), 0% (**2**), 0% (**1**)  **… beyond math and science** 24% (**5**), 40% (**4**), 22% (**3**), 11% (**2**), 8.6% (**1**)  **Core curriculum in your major** 65.2% (**5**), 20% (**4**), 13.04% (**3**), 0.0% (**2**), 0.0% (**1**)  **Technical electives** 63.04% (**5**), 30.4% (**4**), 11.0% (**3**), 0.0% (**2**), 0% **(1**)  **Senior Design Project** 41.3% (**5**), 26% (**4**), 13% (**3**), 4.0% (**2**), 13% (**1**)  (Note that this is a subjective measure, hence it is not used as part of our formal study. Nevertheless, this question maps onto SI VII indirectly, and to indirectly onto PEO “*satisfaction with the decision to further their education*”) |
| **32. How would you rate UCR's effectiveness in preparing you in the following areas?**  (2008)  **Very satisfied- 5 4 3 2 1- Not satisfied**   * **Basic math and science** 50% (**5**), 43% (**4**), 0.0% (**3**), 7% (**2**), 0% (**1**) * **… beyond math and science** 42% (**5**), 50% (**4**), 7.0% (**3**), 14% (**2**), 0% (**1**) * **Core curriculum in your major** 21% (**5**), 50% (**4**), 28% (**3**), 0.0% (**2**), 0% (**1**) * **Technical electives** 28% (**5**), 57% (**4**), 14% (**3**), 0.0% (**2**), 0% **(1**) * **Senior Design Project** 36% (**5**), 36% (**4**), 21% (**3**), 7% (**2**), 0% (**1**)   (2012)   * **Basic math and science** 57.14% (**5**), 26.53% (**4**), 14.28% (**3**), 0% (**2**), 2% (**1**) * **… beyond math and science** 45.83% (**5**), 29.16% (**4**), 12.5% (**3**), 8.3% (**2**), 4.15% (**1**) * **Core curriculum in your major** 59.2% (**5**), 20.4% (**4**), 12.24% (**3**), 4.0% (**2**), 4.0% (**1**) * **Technical electives** 55% (**5**), 22% (**4**), 14% (**3**), 4% (**2**), 4% **(1**) * **Senior Design Project** 46.6% (**5**), 17.7% (**4**), 13.3% (**3**), 6% (**2**), 15.5% (**1**)   (Note that this is a subjective measure, hence it is not used as part of our formal ABET study. Nevertheless, this question maps onto SI VII *indirectly*, and to *indirectly* onto PEO “*satisfaction with the decision to further their education*”) |
| **33. If you participated in research as an undergraduate, how much did that contribute to your preparation for further pursuits?**  **Highly positive - 5 4 3 2 1- Highly Negative**  (2008) 25.0% (2)50.0% (4)25.0% (2)0.0% (0)0.0% (0)  (2012) 45.45% (15)27.27% (9)15.15% (5)9.0% (3)3.0% (1)  (This question maps to SO IV, II and VIII directly, and to *indirectly* onto PEO “*success in post-graduation studies*” ) |
| **34. Our program is designed to enable a Bourns College of Engineering alumnus to be successful either in pursuing a higher degree or in starting a career in science/engineering or a related field. Based on your experience, what comments do you have on our program and our objectives?**  (This is a generic “catch all” question, the answers to which may have implications for any and all ABET criteria and PEOs)  (2008)  “*I think the program still need some work, particularly in the realm of professor's seriousness about training good engineers. In order to have high passing grades and focus on their research, I think professors are too quick to acquiesce to student demands for easier tests, curves, and less homework*.”  “*It needs to be much, much harder to prepare them for the real world and so employers will value UCR candidates more highly*.”  “*Great experience, has literally given my life a new focus and meaning, I highly recommend to any who ask*.”  (2012)  *“I feel that the CS department provided up to date and relevant classes in today's world of fast moving technology. As a CE major, I felt that EE classes hadn't been revised in a decade, labs were unorganized, and TAs unmotivated or unable to speak English. The CS department should strive to work with EE on these joint courses to make them more relevant”*  *“The program did its job.”*  *“It is successful”*  *“I am a patent attorney that specializes in computer software/engineering/internet based technology. While not a career in engineering, an engineering/computer science degree is required”* |

This survey was discussed at length at a faculty meeting on May 9th 2012. Several issues were identified, and a sub-committee was assigned to investigate these and report on them at the Fall 2012 faculty retreat. These issues are detailed in the minutes of the meeting, which are available for inspection.

1. **Student Outcomes**

We consider the Student Outcomes at multiple levels of granularity and at multiple time scales. The core tool we use to do this called the **Student Outcomes Binder**. However over the years it became known colloquially as the **ABET Binder**, and to avoid confusion we will simply use this shorter term below.

The ABET Binder is a physical and logical record of each course offering. The binders for a given quarter are color coded for ease of reference. For example Fall 2011 Binders are yellow etc. They are kept in a secure room, and they are available for inspection by ABET evaluators during sites visits (or at any other time).

The binders are created by the TAs for the offering, under the supervision of the course instructor. The TAs are trained to create the binders in the first two weeks of CS 302 (Apprentice Teaching), which is taught by the ABET coordinator. All TAs must enroll in CS 302 for every quarter in which they TA. In the rare cases in which there is no TA for an offering, the course instructor creates the binder. This situation almost never happens. They only way it can happen is if a class has such a low enrollment that it would normally be canceled, but the Chair decides to offer the class anyway, as a service to the students who need that particular class.

At the end of each quarter, the TAs for each offering must present their binder to the ABET coordinator and have it inspected and “signed off” on. They cannot get credit for CS 302 unless the binder is complete. Thus over the last four years we have had 100% compliance, and the ABET binders form a detailed and carefully annotated archive which we use to assess Student Outcomes (details below).

The ABET binder contains:

* **Course Information**: This includes the full course syllabus, name of the professor and teaching assistants.
* **Handouts**: A copy of every item *physically* handed from the professor to the students, including notes, hard copies of slides, copies of newspaper articles etc. In some cases these may be non-paper items, such as a “fifteen puzzle” used for *CS170* *Introduction to Artificial Intelligence*. For non-physical “objects” such as a suggested URL, the binder either includes the full text/screen dump of the contents of the URL, or in the case of a large resource, just the link and a description of the content.
* **Exams**: A copy of the final exam used in the class, annotated and cross-referenced by its relevance to the student’s outcomes (this is described in great detail below). For some offerings, the midterm(s) are also annotated and recorded in this fashion. *The binder also includes a spreadsheet containing the student’s performance on the exam at a question by question level.* It is this information that allows us to do a fine-grained assessment of Student Outcomes. Note that this paper copy is just for backup, the detailed analysis of this data is done electronically.
* **Quizzes**: A copy of every quiz given (if any), together with one annotated example of “A”, “B” and “C” quality work handed in by (randomly chosen) students. For quizzes that have a binary score, an annotated example of “Credit” and “No-Credit” are kept. Since in most cases the students quizzes are returned to them as a study aid, the binder contains a *copy*.
* **Project**: A copy of every project assigned (if any), together with one annotated example of “A”, “B” and “C” quality work handed in by (randomly chosen) students. If the project involved so much coding that a print-out is not feasible, a CD-Rom or USB stick is included.
* **Homework**: A copy of every homework assigned (if any), together with one annotated example of “A”, “B” and “C” quality work handed in by (randomly chosen) students.
* **Labs**: A copy of every lab assigned (if any), together with one annotated example of “A”, “B” and “C” quality work handed in by (randomly chosen) students.
* **Special notes**: This section is optional, but can be used to make note of any unusual occurrences during the quarter. For example “*Midterm was canceled due to flooding, grade weight on final was increased to 30%*”, or “*Dr. Smith had medical problems in week 6 and, Dr. Keogh finished teaching the class*” etc.

The ABET binders form a tangible, permanent and easy to understand record of every course offering. Faculty members teaching a class for the first time are encouraged to view the relevant binders from previous offerings. When discussing possible changes to a course at a faculty meeting, the relevant binders are brought to the faculty meetings for reference.

**4.B.1 Quantitative Analysis of Student Outcomes**

Our department’s continuous improvement process attempts to optimize two goals. The first is performance on the *course objectives*, which specify the concepts that a specific course strives to teach its students. For example, the introductory CS10 course has “*Use variable to store computer program data*” as one of its course objectives. The second goal is to optimize the performance on the higher level Student Outcomes. The relationships between the course objectives and the Student Outcomes are critical in our assessment process, and have been the subject of significant discussion and adjustment over the years. Although the department largely converged on these mappings by 2005/06, we occasionally revisit the mappings during course revisions and as part of the process for revision of the Student Outcomes/Program Educational Objectives (c.f. Section 2.E)

While *all* these course objectives, with their mapping to our student outcomes are detailed in Appendix A, below we show a random representative example for concreteness. Table 17 shows the mapping of course objectives to Student Outcomes for *CS161L: Laboratory in Design and Architecture of Computer Systems*.

Table 17: Course Objectives CS161L: Laboratory in Design and Architecture of Computer Systems

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Objective to Student Outcome Matrix | | | | | | | | | | | | | |
| Objective Addresses Outcome: 1-slightly 2-moderately 3-substantially | | | | | | | | | | | | | |
| **Outcome Related Learning Objectives** | I | II | III | IV | V | VI | VII | VIII | IX | X | XI | XII | XIII |
| Understanding of computer arithmetic by (1) Design and implementation of an ALU and (2) Implementation of complex arithmetic algorithms in software. | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 0 | 0 |
| Understanding of operation of a CPU by (1) Design and implementation of a data-path and (2) Design and implementation of a the control unit both for the MIPS architecture | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 0 | 0 |
| Understanding of operation of a cache memory by designing and writing a cache-simulator program in C/C++ | 3 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 2 | 0 | 0 |
| Familiarity with the cycle-level simulation of a complex computer architectures | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 |
| Understanding of data-paths via a hands on introduction to data-paths | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 3 | 0 | 0 |

The last major revision of these mapping happened in 2008/09, using the following process. For each course, a committee was formed consisting of everyone that had ever taught the course, everyone that *might* teach it in the future (i.e., new hires in that area) and any interested party. The committee drafted the mapping, after viewing the previous syllabi, teaching materials, and examples of graded instruments (exams/quizzes/homeworks/projects). The draft mappings were presented and discussed at several faculty meetings in the fall of 2008 and voted on by the full faculty.

Below we provide a detailed work explanation of how we analyze the student’s achievements on the final (and sometimes, *midterm*) for both *coverage* and *performance* on course objectives.

* *Coverage* measures the extent to which we are teaching and testing the course objectives.
* *Performance* measures the student’s knowledge of the course objectives.

Note that our decision to measure these metrics on only the final (and sometimes, *midterm*) exam was a conscious choice made at the faculty retreat in fall 2007. The final and midterm exams are always conducted in a controlled environment under the professor’s direct supervision, with 100% attendance, and thus avoid the difficulty of confounding factors such as absent students (quizzes), or students cheating (homeworks) etc.

In Table 18 below, the term course objective has been shortened to *objective*, and the term Student Outcome has been shortened to *outcome*. Table 18 shows a running example we will use to explain our process. The Student Outcomes are shown as **A**, **B**, & **C** in the table.

Table 18: A worked example of how we determine *coverage* and *performance* on course objectives and Student Outcomes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Exam Scores** | Q1 | Q2 |  |  |  |
| John Doe | 1 | 1 |  |  |  |
| Jane Smith | 0 | 1 |  |  |  |
| Normalized Question Performance | 0.5 | 1 |  |  |  |
|  |  |  |  |  |  |
| **Question-To-Objective** | Q1 | Q2 |  | Objective Performance | Objective Coverage |
| Objective 1 | 3 | 3 |  | 0.75 | 1 |
| Objective 2 | 0 | 3 |  | 1 | 0.5 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Objective-To-Outcome** | **A** | **B** | **C** |  |  |
| Objective 1 | 3 | 3 | 3 |  |  |
| Objective 2 | 0 | 0 | 0 |  |  |
|  |  |  |  |  |  |
| Outcome Performance | 0.75 | 0.75 | 0.75 |  |  |
| Outcome Coverage | 0.5 | 0.5 | 0.5 |  |  |

The numbers in white cells are entered by faculty for each course offering; whereas all the numbers in gray cells are calculated (by automatic scripts) as part of our evaluation process, two weeks after finals week.

The performance data here is being calculated from two students’ scores on a single, two-question exam, as seen in the “Exam Scores” table. In our running example, John Doe answered two questions perfectly (1 out of 1 on both questions), whereas Jane Smith got zero credit on the first question, but aced the second question.

The “Question-To-Objective” table gives a weight to each objective-question pair, which states how related the question subject is to the objective. These weights are determined and recorded by the course instructor, as he/she writes the exam.

The “Objective-To-Outcome” table gives a weight (i.e. mapping) to each objective–outcome pair. Recall (as discussed above) these weights are relatively static, and have been essentially unchanged for 5 years. This is important, since it allows us (with appropriate statistical caution) to compare offerings of the same course offered in different quarters, and to plot and consider trends.

We wish to calculate how well a course covers its own objectives as well as the Student Outcomes. We also calculate how well students of the course perform with respect to each objective/outcome. The steps to produce the derived data are as follows.

1. **Normalized Question Performance**: Each question’s score is normalized by dividing it by the maximum possible number of points for that question. Then the average score across students is found for each exam question. In the example above, in the “Exam Scores” table, the max question point value possible is 1.
2. **Objective Performance**: The relevance of each exam question to each course objective is given as input, as seen in the “Question-To-Objective”. To determine the performance of an objective we first multiply the Normalized Question Performance of a question by the objective-to-question weight for that objective, for each exam question. We then divide the sum of the sum of the objective-to-question weights for that objective. Table 19 shows an example of how this formula is used for the running example. Note that the performance for objective 2 is perfect since question 1, which had imperfect performance, is not relevant to objective 2 and thus not used in its calculation. Thus, if the faculty reviewed this data they would be happy to discover perfect performance on Objective 1, but would worry about the relatively poor performance on Objective 2.

Table 19: The Calculation of the Objective Performance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Exam Scores** | Q1 | Q2 |  |  |
| Normalized Question Performance | 0.5 | 1 |  |  |
|  |  |  |  |  |
| **Question-To-Objective** | Q1 | Q2 |  | Objective Performance |
| Objective 1 | 3 | 3 |  | [(0.5\*3) + (1\*3)] / (3+3) = 0.75 |
| Objective 2 | 0 | 3 |  | [(1\*3) + (1\*3)] / (0+3) = 1 |

1. **Objective Coverage**: Knowing the *performance* on the course objectives is of little use without an understanding of the *coverage*. If nothing else, with very low coverage, the performance results are likely to be unreliable. To determine the coverage of an objective we first sum the question-to-objective weights for that objective. We then divide by the maximum possible coverage value, which is the number of questions multiplied by the max possible weight, which in this example is 3. Table 20 shows the relevant calculations for our running example.

Table 20: The Calculation of the Objective Coverage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question-To-Objective** | Q1 | Q2 |  | Objective Coverage |
| Objective 1 | 3 | 3 |  | (3+3) / (2\*3) = 1 |
| Objective 2 | 0 | 3 |  | (0+3) / (2\*3) = 0.5 |

1. **Outcome Performance**: To determine the performance of an outcome, we first multiple each objective’s performance value by its objective-to-outcome weight for that outcome, and sum over all objectives. We then divide by the maxium possible achievable value, which is just the sum of all of the objective-to-outcome weights for that outcome. Table 21 shows an example of this calculation. Note that since only objective 1 relates to each outcome (**A**, **B**, **C**), the outcome performance draws only upon objective 1’s performance, and thus are all the same.

Table 21: The Calculation of the Outcome Performance

|  |  |
| --- | --- |
| **Question-To-Objective** | Objective Performance |
| Objective 1 | 0.75 |
| Objective 2 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Objective-To-Outcome** | **A** | **B** | **C** |
| Objective 1 | 3 | 3 | 3 |
| Objective 2 | 0 | 0 | 0 |
|  |  |  |  |
| Outcome Performance | [(0.75\*3) + (1\*0)] / (3+0) = 0.75 | 0.75 | 0.75 |

1. **Objective Coverage**: To determine the coverage of an outcome, we first multiply each objective’s coverage value by its objective-to-outcome weight for that outcome, and sum over all objectives. We then divide by the maximum possible coverage value, which is the number of objectives multiplied by the max possible weight, which in this example is 3. In Table 22 each outcome is fully covered by objective 1, which is in turn fully covered on each question. However since each outcome is missing coverage by objective 2, their final coverage is only 0.5.

Table 22: The Calculation of the Objective Coverage

|  |  |
| --- | --- |
| **Question-To-Objective** | Objective Coverage |
| Objective 1 | 1 |
| Objective 2 | 0.5 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Objective-To-Outcome** | **A** | **B** | **C** |
| Objective 1 | 3 | 3 | 3 |
| Objective 2 | 0 | 0 | 0 |
|  |  |  |  |
| Outcome Coverage | [(1\*3) + (0.5\*0)] / (3\*2) = 0.5 | 0.5 | 0.5 |

1. **Multiple Exams**: Our method can also handle the case of multiple exams. The summation of values is taken over all exams and then metrics are computed, rather than metrics being computed per exam and then averaged by number of exams. Below you can see a simple example with one student, one question, and two exams. The “*Exam ‘x’ Objective Performance*” tables show the performance metric for each exam individually. The “False Objective 1 Performance” table shows the result if the per-exam values are simply averaged. The problem with this is that the relative weight of the question-to-objective is lost between the same objective across exams. The “*Actual Objective 1 Performance*” shows the correct way to calculate across exams, where the summations are carried out across exams, and the objective-to-question weighting is preserved.

Table 23: Combining Results of Two Exams

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Exam 1** | Q1 |  |  |  |  |
| John Doe | 0 |  |  |  |  |
|  |  |  | Exam 1 Objective Performance |  |  |
| Objective 1 | 1 |  | (0\*1) / 1 = 0 |  | ABET **Course** Metric |
|  |  |  |  |  | False Objective 1 Performance |
|  |  |  |  |  | (0 + 1) / 2 = 0 .5 |
| **Exam 2** | Q1 |  |  |  |  |
| John Doe | 1 |  |  |  | Actual Objective 1 Performance |
|  |  |  | Exam 2 Objective Performance |  | (0\*1 + 1\*3) / (1+3) = 0 .75 |
| Objective 1 | 3 |  | (1\*3) / 3 = 1 |  |  |

1. **Zero Coverage Case**: Using the objective/outcome performance calculation given before would give you a value of 0 performance if the coverage was also 0. The problem is that in this case, because the objective did not relate to any questions, we do not have any information as to how the students would perform on that objective. We handle this as a special case, as indicated by the -1 performance value for objective 1 seen below. When summing the product of each objective performance / objective-to-outcome pair to find outcome performance, we ignore any objectives which have -1 performance. The outcome’s performance, just like objective performance, is given a -1 if that outcome’s coverage is 0.

Table 24: How the Pathological Zero Coverage Case is Handled

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Question-To-Objective** | Q1 | Q2 |  | Objective Performance | Objective Coverage |
| Objective 1 | 3 | 3 |  | xxx | 1 |
| Objective 2 | 0 | 3 |  | -1 | 0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Objective-To-Outcome** | **A** | **B** | **D** |  |  |
| Objective 1 | 3 | 0 | 0 |  |  |
| Objective 2 | 3 | 0 | 3 |  |  |
|  |  |  |  |  |  |
| Outcome Performance | xxx | -1 | -1 |  |  |
| Outcome Coverage | 0.5 | 0 | 0 |  |  |

This section has been very long and technical. However the reader is now in a position to appreciate the following section, where we show how we use these formulas to produce intuitive visual summaries of student outcomes at multiple granularities, in order to support and direct our efforts in continuous improvement.

*4.B.1.a Course Reports*

Using the analytical algorithms discussed in the previous section, for *every* quarter, for *all* course offerings, we produce a visual report summarizing the *coverage* and *performance* on the course outcomes, together with the *coverage* and *performance* on the PEOs. Figure 15 shows a randomly chosen example.

|  |
| --- |
|  |

Figure 15: A Sample Course Report, from the spring 2010 offering of CS141: *Intermediate Data Structures and Algorithms*

This report is produced within a week of the final grades being submitted, and is then sent to the course instructor, who compares it with the reports from previous offerings. The course instructor is encouraged to annotate the report. For example, a recent report was annotated: “*The earlier introduction to the Relational Model does seem to have had the desired effect, however, the student’s performance on query evaluation and the internals of query optimization seems to have stalled*... ”. The annotated reports are archived with the ABET binders (C.F Section 4.B), and are discussed at faculty meetings, and at the annual faculty retreat.

More critically, an instructor teaching an offering of a course is required to view the course report for at least the most recent offering (*all* reports are available if desired) and to contact the previous instructor (if a different person) to be briefed on the course. It is this step that is codified in Step 1 of Table 15, *Data Collection for Course Assessment*.

*4.B.1.a Course Difference Reports*

The course reports shown in the previous section allow the instructor an intuitive review of the coverage and performance on course objectives. Suppose an instructor spots an area of concern, perhaps the course is not covering an objective adequately (alternatively, an objective is being *taught*, but not adequately *tested*), or an objective is being covered, but the students are testing poorly on it. As shown in the “inner loop” of Table 15, the instructor in consultation with his/her colleagues that have experience/interest in teaching the class, and the undergraduate instruction committee, will attempt to find a remedy for the next course offering. How can we tell if that remedy is successful? For this purpose we generate a special *course difference report*, which uses the formula introduced in Section 4.B.1 to produce a side-by-side comparison of two offerings. Figure 16 shows an example of such a report, contrasting the Winter/Spring 2010 offerings of *CS010: Introduction to Computer Science*.

|  |
| --- |
|  |

Figure 16: A Sample Course Difference Report

The courses in question were taught by Kris Miller. Below are his unedited notes on this course difference report:

*One reason I believe scores for almost all course objectives went up in Spring 2010 versus Winter 2010 is that we switched from teaching the students to use the editor emacs to the IDE Code::Blocks starting with the Spring 2010 quarter. Brian, Victor, and I had actually been discussing this change for several quarters. We saw that students struggled mightily with tool shock the first couple of weeks. Many students in fact never really became comfortable with emacs. We felt that if we could find a program editor that more closely resembled the types of text editors students were used to coming from a Windows background this would reduce some of the tool shock, allowing the students to more quickly focus on the programming concepts rather than the programming tools.*

*The TAs reported after the first week of labs that they noticed the students were definitely struggling less with the programming tools and the course reports appear to support the case that this had a beneficial effect on the entire quarter.*

Note that the detailed information captured in the course reports and course difference reports is useless unless it is examined and acted on. As hinted at in Figure 17 the *entire* faculty does examine these reports, together with the contents of the ABET binder, at faculty meetings.

|  |
| --- |
|  |

Figure : ABET course reports are regularly examined at faculty meetings when discussing undergraduate education, curriculum changes, effectiveness of teaching (for promotion and tenure cases) etc.

**Expected level of attainment for achievement of all Student Outcomes**

The CS Steering Committee has decided on an expected level of attainment for achievement of all Student Outcomes to be 70%. The rationale for this number is:

* The attainment numbers will invariably include students that choose to quit the major. As these are generally the poorest performing students, they are “*pulling down*” the averages. Thus the attainments numbers we see in our reports are in essence lower bounds for the students that actually graduate.
* The ABET committee has consulted with other UC campuses with similar programs, and our chosen value is not significantly different from theirs.
* We have done significant data analyses on the amount of uncertainly in our measurement system (and how to reduce it). For example he has extensively studied the variance in attainments when everything else is fixed (Same instructor, same course, approximately same class size, same exam questions (with values changed and other anti-cheating devices)). Given the amount of inevitable uncertainly in our measurement system, a threshold higher than 75% is very likely to produce many Type II errors (we appear to miss our attainment goals, but we actually are meeting them). Our 70% goal, is much less likely to produce such errors, but still holds us to a high level of attainment.

We revisit these expected levels of attainment for achievement at least once a year, the last occasion they were discussed was May 23th 2012.

In Table 25 we show the results of aggregating the *coverage* of student outcomes at a quarter by quarter level. In the left side of the table we show a binary table, which simply tells us if each student outcome was covered or not in a particular quarter. Note that in most quarters all outcomes are covered, but it is possible that in a given quarter we may not cover all outcomes, depending on the courses offered that quarter. In the right side of the table we show the *relative* coverage (that is to say, the columns sum to one).

Table 25: The Student Outcomes Coverage Aggregated for each Quarter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Binary Outcome Coverage | | | | | | Normalized Outcome Coverage | | | | | | | | | SO | *Win10* | *Spr10* | *Fall10* | *Win11* | *Spr11* | *Fall11* | | *Win10* | *Spr10* | *Fall10* | *Win11* | *Spr11* | *Fall11* | | *I* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.27 | 0.20 | 0.17 | 0.17 | 0.24 | 0.23 | | *II* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.06 | 0.03 | 0.05 | 0.05 | 0.05 | 0.14 | | *III* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.11 | 0.08 | 0.12 | 0.12 | 0.09 | 0.17 | | *IV* | 0 | 1 | 1 | 1 | 1 | 0 |  | 0.00 | 0.17 | 0.16 | 0.16 | 0.06 | 0.00 | | *V* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.02 | 0.03 | 0.02 | 0.02 | 0.02 | 0.01 | | *VI* | 1 | 1 | 1 | 1 | 1 | 0 |  | 0.03 | 0.01 | 0.00 | 0.00 | 0.03 | 0.00 | | *VII* | 0 | 1 | 1 | 1 | 1 | 0 |  | 0.00 | 0.06 | 0.04 | 0.04 | 0.01 | 0.00 | | *VIII* | 0 | 1 | 1 | 1 | 1 | 0 |  | 0.00 | 0.02 | 0.05 | 0.05 | 0.03 | 0.00 | | *IX* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.44 | 0.29 | 0.24 | 0.24 | 0.37 | 0.32 | | *X* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.15 | 0.10 | 0.11 | 0.11 | 0.13 | 0.18 | | *XI* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.28 | 0.19 | 0.18 | 0.18 | 0.23 | 0.25 | | *XII* | 1 | 1 | 1 | 1 | 1 | 1 |  | 0.07 | 0.05 | 0.10 | 0.10 | 0.05 | 0.13 | | *XIII* | 0 | 1 | 1 | 1 | 1 | 0 |  | 0.00 | 0.05 | 0.04 | 0.04 | 0.04 | 0.00 | |

This table was last shown to the faculty on May 2nd 2012. The faculty noted that relatively sparse *coverage* of some outcomes (i.e. IV, VII, XIII) makes the determination of performance of the outcomes (see below) more difficult. It was decided that the ABET committee and the undergraduate education committee investigate the issue and report to faculty at the Faculty retreat in Fall 2012.

In Table 26 we show the results of aggregating the *performance* of student outcomes at a quarter by quarter level. Note the handful of missing entries correspond to outcomes that were not tested in that quarter (at least by an instrument that was recorded in our system, the outcomes were almost certainly measured by other instruments used to determine the students grade).

Table 26: The Student Outcomes Performance Aggregated for each Quarter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Outcome Performance | | | | |  |  |  | Max Outcome Performance | | | | | | SO | Win10 | Spr10 | Fall10 | Win11 | Spr11 | Fall11 | | Win10 | Spr10 | Fall10 | Win11 | Spr11 | Fall11 | | *I* | 0.62 | 0.64 | 0.69 | 0.69 | 0.69 | 0.72 |  | 0.79 | 0.96 | 0.82 | 0.82 | 0.85 | 0.78 | | *II* | 0.58 | 0.69 | 0.67 | 0.67 | 0.56 | 0.63 |  | 0.73 | 0.96 | 0.81 | 0.81 | 0.78 | 0.63 | | *III* | 0.58 | 0.69 | 0.70 | 0.70 | 0.67 | 0.74 |  | 0.71 | 0.96 | 0.82 | 0.82 | 0.84 | 0.83 | | *IV* |  | 0.62 | 0.75 | 0.75 | 0.75 |  |  |  | 0.62 | 0.75 | 0.75 | 0.75 |  | | *V* | 0.55 | 0.64 | 0.75 | 0.75 | 0.61 | 0.89 |  | 0.62 | 0.80 | 0.78 | 0.78 | 0.85 | 0.89 | | *VI* | 0.41 | 0.66 | 0.78 | 0.78 | 0.46 |  |  | 0.41 | 0.66 | 0.78 | 0.78 | 0.68 |  | | *VII* |  | 0.81 | 0.55 | 0.55 | 0.59 |  |  |  | 0.96 | 0.57 | 0.57 | 0.59 |  | | *VIII* |  | 0.96 | 0.54 | 0.54 | 0.70 |  |  |  | 0.96 | 0.54 | 0.54 | 0.70 |  | | *IX* | 0.62 | 0.63 | 0.69 | 0.69 | 0.68 | 0.72 |  | 0.79 | 0.96 | 0.81 | 0.81 | 0.85 | 0.81 | | *X* | 0.59 | 0.67 | 0.69 | 0.69 | 0.64 | 0.70 |  | 0.74 | 0.96 | 0.82 | 0.82 | 0.82 | 0.75 | | *XI* | 0.60 | 0.66 | 0.70 | 0.70 | 0.68 | 0.73 |  | 0.75 | 0.96 | 0.82 | 0.82 | 0.85 | 0.82 | | *XII* | 0.51 | 0.63 | 0.71 | 0.71 | 0.68 | 0.67 |  | 0.79 | 0.96 | 0.83 | 0.83 | 0.85 | 0.73 | | *XIII* |  | 0.80 | 0.63 | 0.63 | 0.63 |  |  |  | 0.96 | 0.78 | 0.78 | 0.63 |  | |

In the next section we give numerous examples of *continuous improvement*. Note that most of these examples were driven by a need noted in the course reports (among other methods), and almost all improvements are measured by changes in the outcomes in the course reports (among other ways).

Note that for clarity of presentation here we have omitted values or graphics that show the *variance* of these measurements. However as scientists we recognize the importance of consider variance while assessing the potential significance of any apparent changes in performance. This extracted information is available and considered at every step.

1. **Continuous Improvement**

Without claiming to be exhaustive, below we list some *representative* examples of how the results of evaluation processes for the program educational objectives and the student outcomes and other available information have been used as input in the continuous improvement of the program.

Note that many of these changes are rather small; we feel that we had converged on an excellent program by 2006. Nevertheless the faculty is encouraged to, and rewarded for, fine-tuning any aspect of the curriculum.

**Improvements to AI offerings based on discussions with constituents (Outer Loop, Table 15)**

In 2011/2012, we significantly restructured our artificial intelligence offerings. Our inspiration to do so came from a variety of sources. The most significant of these were meetings between the two faculty most responsible for AI (Dr. Shelton and Dr. Keogh), meetings with the Board of Advisors, and Dr. Keogh’s meetings with several student employers, including Microsoft, ESRI and ISCA technologies (each of these companies have hired at least four of our students in the last five years). Moreover, Dr. Shelton spoke at length with instructors at other UC campuses about their experiences with undergraduate AI courses. Finally, Dr. Keogh interviewed several alumni who had taken CS 170: Intro to AI with him, and had gone on to work in a position that might reasonably use these skills (including Shruti Kasetty, Microsoft, Isaac Espinoz, PricewaterhouseCoopers etc).

Our findings from these constituents included:

* **Need**: We needed to offer an *information retrieval* course, as a very significant fraction of our students end up working in a related field (text retrieval at Microsoft Bing, Spatiotemporal information retrieval at ESRI, multimedia retrieval at ISCA technologies etc.) **Action**: We designed a new course (cf. Section 6.E), CS 172: *Introduction to Information Retrieval*. Moreover, when making hiring decisions in the 2010/11 cycle, the need for an instructor for this course was weighted. While (at least) Dr. Shelton, Dr. Keogh and Dr. Tsotras can teach this class, it was felt it would be good to have a specialist help develop the curriculum, and teach the first offering. In fact, we did hire Dr. Hristidis, an information retrieval specialist, and he did lead the charge in developing the curriculum, and he is teaching the first offering in spring 2012.
* **Need**: Our BOA questioned the need for us to have CS 171: *Introduction to Expert Systems* on the books, as this is now seen as a dated and unused technology, and many of its applications have been subsumed into other areas (Bayesian Networks etc). As it happens, that course had not been offered in over a decade. **Action**: In winter 2010, a committee of AI professors met to discuss this, and they proposed to the faculty that this course be removed from the curriculum. This was unanimously approved by the faculty.
* **Need**: Our BOA and (some of) our employers suggested that our AI class focus less on “classic AI” (search, logic, constraint satisfaction etc), and more on modern statistical/probabilistic learning techniques, as these are more pragmatic skills for companies such as Facebook, Microsoft etc. However, the faculty felt that the “classic AI” material was important to keep because it offered an opportunity for projects where students could both gain significant experience in coding, and formally prove certain properties of their programs (completeness, optimality etc). **Action**: We restructured the course offerings. Beginning in 2012/2013, CS 170 will be a “classic AI” course on search, logic, constraint satisfaction, and the like. CS 171 will be a “machine learning / data mining” course that will consider statistical/probabilistic learning techniques. We feel these changes update the curriculum to a more modern standard and allow better coverage of the relevant topics. As we roll out the first offerings of these courses in 2012/2013 a committee consisting of Dr. Shelton, Dr. Keogh, Dr. Hristidis, and Dr. Tsotras will carefully evaluate the effects of these changes, both by using the classic measure of our course reports, and by more explicit surveying of the students that take the classes.

**Improvements to CS180 based on viewing course reports (Inner Loop, Table 15)**

In winter 2010 Dr. Neamtiu taught CS 180: *Introduction to Software Engineering*. The course report for the offering is shown in Figure 18.

|  |  |
| --- | --- |
| Notes: This is necessarily a low-resolution view of the course report. The original high resolution version, together with all the original raw data is available for inspection.  This report only reflects the course final, which was not cumulative. The midterm *does* reflect coverage on course objective 3. Nevertheless, based on a review of this report with the undergrad committee Dr. Neamtiu has decided that future offerings will have a cumulative exam, in order to allow a more meaningful comparison between different offerings of the class. |  |
| **Course Objectives for CS180**  1: Provide students with a broad overview of software engineering, covering all phases of the software life cycle and a variety of software process models.  2: Provide students with a variety of techniques for requirements analysis, architectural and detailed design, validation, and verification, as well as planning and management.  3: Provide Students with practical experience in applying such techniques by producing a (small) software product throughout the course and handling certain documents as required in a “classical” (i.e., non-agile) setting as milestones.  4: Allow students to gain experience in scheduling and manage their projects using hands-off approach to team formation, planning, and form of deliverables. |

Figure 18: The Course Report for CS180-Winter 2010

Dr. Neamtiu noted several things from the course report which prompted him to make changes.

* **Need**: First, he noted that we were poorly assessing how well the students understand the ethics of software engineering and the societal implications of software errors. Moreover, from what little assessment we were doing the students’ performance was lacking. **Action**: Dr. Neamtiu added contents to the lecture material to illustrate the consequence of software-induced disasters: the explosion of the Ariane 5 rocket, the 2003 Northeastern Blackout, the Marc Climate Orbiter explosion, avionics software. Also he added the following essay-like question on the final exam: “*In engineering disciplines such as civil engineering, a product (e.g., bridge) failure might lead to the product designers and makers being prosecuted. However, in software engineering, product designers and makers are not only not held responsible, but the product itself usually comes with no warranty whatsoever, even though, as illustrated in class, software failures puts lives and livelihoods at risk. Why do you think software engineering is held to a lower responsibility standard than other engineering disciplines?*”.
* **Need**: As per Course Objective #3, practical experience in building a small software product and handling documents in a milestone-setting is deemed important. Moreover, at the department’s Board of Advisors meeting in March 2010, leaders from industry, e.g., Yahoo! Research have noted that one of the best improvements we can make to our students’ preparation was to turn them into *software engineers*, rather than *programmers*. Briefly, the difference between the two categories is that programming essentially an individual activity, whereas software engineering involved multi-person construction of multi-version software. **Action**: In CS 180, for the project part of the course, Dr. Neamtiu has moved to an approach that emphasized flexibility in team formation and implementation strategy, while adhering to strict documentation and schedule guidelines; this approach has the role of exposing students to realistic software development practices, as well as issues and solution that appear in the development of large projects in a multi-person team (this also addresses Course Objective IV). Moreover, projects change each quarter and topics are highly relevant to what students will work on after graduation. For example, recent team projects included implementing “lite” versions of LinkedIn, Facebook, YouTube, Twitter. Other projects involved developing Android and iPhone applications for augmented reality and time management. Finally, students have to construct their project in a succession of two milestones, to simulate the incremental delivery approach standard in real-world development. Projects are graded on requirement completion as well as documentation and a project demo presentation. The approach has already started to bear fruit. Apart from observing numerical improvement on course reports, Dr. Neamtiu has personally followed up by contacting both alumni, finding that they indicate they have been able to find jobs and internships based on the large, team-oriented, highly-relevant-to-employers project in the class, and student employers, who report increased satisfaction with our students’ performance.

**Improvements to CS111 based on viewing course reports (Inner Loop, Table 15)**

Dr. Chrobak frequently teaches *CS111: Discrete Structures*. A sample report for a recent offering of this course is shown in Figure 19.

|  |  |
| --- | --- |
| Notes: This is necessarily a low-resolution view of the course report. The original high resolution version, together with all the original raw data is available for inspection. |  |
| **Course Objectives for CS111**  1: To learn how to use correct mathematical terminology and notation.  2: To learn methods of formal mathematical reasoning and proof techniques, including proofs by contradiction and by induction.  3: To learn how to model real-life problems using discrete mathematical structures: sets, sequences, combinations, permutations, graphs, trees, relations, and algebraic structures.  4: To master the concept of asymptotic notation and its application to estimating running time of various algorithms.  5: To learn techniques for solving recurrence equations, and their applications to counting and to analyze the complexity of divide-and-conquer algorithms.  6: To learn the basic concepts in graph theory including connectivity, cycles, planarity, coloring. |

Figure 19: The Course Report for CS111-Winter 2010

Dr. Chrobak noted several things from the course reports which prompted him to make changes to the course.

* **Need**: Dr. Chrobak noticed that students were scoring relatively poorly on outcome 1, his annotation of course reports included text such as “*I noticed that many students, even those with solid understanding of the material, had difficulty presenting their solutions in a clear and rigorous fashion.”*. **Action**: After discussion with the other faculty that teach this course, and with the undergraduate education committee, Dr. Chrobak took action to address this problem: “*To address this, I introduced several changes to the way CS111 is taught. I provide students with strict guidelines for preparing homework assignments. At the beginning of the course students are required to read several articles on writing mathematical text. In lecture, I put more emphasis on rigorous formulations and reasoning. For example, when presenting Kuratowski's theorem “*A graph is planar if and only if it does not contain a subgraph that is a sub-division of K5 or K3,3*”, I would discuss common mistakes, like replacing “*if and only if*” by “*if*” or by “*only if*” weakens the statement, or why writing “*does not contain K5 or K3,3*” (that is, omitting the sub-division part) makes the statement incorrect. Questions to formally state definitions or theorems are now included on tests*.” The results in the most recent offering of the course, are shown in Figure 19, and they strongly suggest that this process is helpful, the outcome 1 scores have risen to be on par with other outcomes.
* **Need**: Dr. Chrobak made changes the syllabus for CS111 to address students' difficulties with the material (after consulting the entire faculty, at a faculty meeting in winter 2010). In earlier offerings of CS111, we covered *generating functions*. That topic caused significant difficulties for the students, partly because it requires solid background in calculus. Dr. Chrobak consulted with our BOA and several employers who note that *generating functions* are of limited educational value to students that major in computer science or computer engineering. **Action**: In more recent offerings of CS111 we now instead cover *counting integer partitions*, a topic that is mathematically equally sophisticated, but does not require background outside of discrete mathematics, and is more relevant to mathematical problems students are likely to encounter in their future study.

**Improvements to curriculum based on alumni surveys (Outer Loop, Table 15)**

Dr. Zordan was, until 2010, the only faculty in computer graphics, and as such anxious to improve the graphics part of our curriculum. He combed over the 2009 alumni survey, visited employers of our students with an interest in graphics (Dreamworks, Sony, THQ, and Electronic Arts.) and did personal one-on-one phone follow up interviews with alumni. Based on these efforts, he noted:

* **Need**: Students interested in video games and graphics as freshmen had no way to pursue their interest until their junior to senior year because there was a long list of requirements before they could take CS130 *Intro to graphics*. Many students seemed discouraged and unsure of their choice to invest several years in a discipline they saw as only peripherally related to their interests. **Action**: Dr. Zordan led the charge in modifying the curriculum to add CS66 *Introduction to Three-Dimensional Digital Modeling* and CS67 *Three-Dimensional Digital Modeling and Animation*, as courses with no prerequisites that teach both about the topic and software as well as programming via a games industry scripting language. These courses are in their first complete run this academic year (11-12) so there are no hard measurements, but we will track to see if these students appear to yield higher retention rates over the average. We anticipate this as a result of their satisfaction and engagement in taking such courses earlier in their undergraduate experience. Anecdotal evidence shows that students appear more confident and satisfied after taking CS66. Their confidence is both in their choice of studies (computer science as a discipline that leads to pursuits in computer game development) as well as their sense of proficiency and experience in their field of choice (as programmers and masters of a suite of software that is used ubiquitously in the game industry).

**Improvements to CS120B based on viewing course reports (Inner Loop, Table 15)**

Dr. Frank Vahid frequently teaches CS120B Introduction to Embedded Systems. Note that this course is listed as EE/CE 120B and is co-taught by EE and CS. This course has lab involving hardware (breadboards, chips, components, wiring, troubleshooting tools) and software (C programming, compilation/debug tools, downloading).

* **Need**: based on a careful analysis of the course reports, and interviews with the students, Dr. Valid noted that on the midterm and homework, students from EE did very well on hardware questions, but less well on software questions, and the opposite was true for students from CE. Afraid that this observation might have been a fluke, Dr. Valid asked other faculty that teach this course, including Dr. Najjar (CE) and Dr. Zhu (EE) to test to see if this was true in their offerings. They confirmed the finding, anecdotally at first, and then numerically using the course reports. Together they hypothesized the following cause: Students work in pairs and share a hardware lab kit that they purchase, as well as a lab PC, as is standard in most universities offering embedded systems courses. However, a drawback of this approach that was observed was that commonly one partner (typically an EE student) would do most of the hardware work and the other partner (typically a CS student) all of the software work, leading to an imbalanced learning experience.

**Action**: Dr Vahid tried several methods to encourage all students to learn both HW and SW elements, such as practical exams. However, this had limited success (as measured by course reports), were hard to manage and anecdotally seem stressful for students. Dr. Vahid takes up the story in a report he wrote as a briefing to the faculty...

*For Winter 2012, we decided to try the approach of requiring every student to have their own lab kit (which is more feasible than 5 years ago due to dropping costs for the kits). Thus, in Fall 2011, we spent about 100 hours upgrading the lab materials with detailed explanations, extensive photographs, and some videos, so that CS people could be successful with learning the hardware kit, and EE people successful with learning the software tools. Then, with each student having their own kit, we had students rotate partners for the first 5 weeks of the course. For the last 5 weeks they could choose their partner. For the final project, each partner did his/her own project, but then 15% of the grade was based on their partner's project and interaction, striking a balance between each student learning independently, and students getting help from a partner and learning to work together. The approach was a success, with students doing equally well on their lab assignments and final projects (as measured by the course reports) even though working more independently and having to learn both hw and sw. Thus we are continuing the one-lab-kit-per-student approach this quarter and will do so from now on. I observed that students were having difficulty on the midterm exam capturing synchronous state machines to specify desired embedded systems program behavior. I also noted that students usually did significantly better on concepts that were done on a homework. The problem with the synch SM concept though is that, even though students did them on homeworks, there was no quick way of them knowing if they got the problem right or wrong (in contrast to other types of problems), and students rarely spend time poring over a solution to compare their answer. I also noted that, although we were teaching this disciplined synchSM method in lectures, students in the lab projects would commonly ignore the method and use ad hoc methods (as is done in most universities) -- I believed that more practice in the synch SM methods and perhaps tools to ease use of the method might remedy the issue. Thus, a colleague of mine at UCI and I decided that students needed a "virtual lab" tool that would capture synch SMs and allow for their simulation. We wrote a proposal to NSF's CCLI program (Course, Curriculum, and Lab Improvement) and obtained a joint grant. We hired two undergrads who spent a year developing a synchSM capture tool and a microcontroller simulator tool, which were integrated with one another, called the RI (Riverside-Irvine) tools. These tools are now a centerpiece of the course, with all homeworks done using the tools, and even the first week of lab spent just learning the tools. Student performance has improved tremendously. Students regularly get A/B scores on exam problems for capturing synchSMs. Furthermore, we were able to extend the course content to emphasize capturing concurrent synchSMs. Student final projects in the last two weeks of the course are now impressive 4-5 concurrent synchSM systems running on a single microcontroller with no operating system support -- projects that could never have been completed several years ago before the tools. As part of the NSF project, we also developed an e-book, and will soon begin advertising the book and tools to other universities.*

Note that Dr Vahid both noted the need for change, and measured the results of the change using (among other tools) the course reports.

**Improvements made by a single faculty member**

In the above example we considered data-driven continuous improvement to *individual courses*. In this section we review data-driven continuous improvements created by an *individual* professor. We consider the work of Dr. Philip Brisk, simply to avoid redundancy (he was not mentioned above), however similar remarks could be made for all our faculty. Below we have pasted Dr. Brisk’s notes verbatim, with no attempt to edit or “polish”.

*RATIONALE: I received a grant from the Cyber Physical Systems (CPS) Program of the US National Science Foundation (CNS-1035603, “System support for generally programmable digital microfluidic biochip devices”). One of the criteria for the successful outcomes of NSF grants is "integration of research with education," but the actual methods employed for integration and evaluation are left unspecified.*

*IMPROVEMENT: I did two things, relating to two different classes that I teach.*

*CS.120B: This applies to the offering of CS.120B in Spring 2011. Every week, I assigned a paper to the students to read based on CPS. I discussed the paper for 10-15 minutes during each week, and supplanted the discussion with videos whenever possible. CPS-related themes included robotics, wearable computing, MS Kinect, and programmable microfluidics, among others. I asked basic questions on the exams, and occasionally on homeworks, to assess whether the students were reading and understanding the papers that were assigned.*

*CS.179J: I created specific senior design projects relating to programmable microfluidic technology, which is the core topic of my grant. In the Spring 2010 offering, one group of three students adopted one of these projects; in the Winter 2012 offering, two groups of four students (eight students in total) adopted these projects.*

*RESULTS:*

*CS.120B: I monitored the performance of students on homework's and exams on the questions relating to CPS and external readings. Initially, the students did not do particularly well; my interpretation was that the students did not take these readings seriously, as they were somewhat decoupled from the core topics covered in the textbook, lecture, and lab; however, student performance improved significantly as the quarter progressed. I noticed that many topics were too advanced for the students to comprehend in detail, even though they were making an effort. The problematic topics were robotics and computer vision (as applied to the MS Kinect). On the bright side, 6 of the top students in the course approached me about doing independent research during the Summer of 2011, and one approached me later (at the end of the Fall 2011 quarter). All performed independent research in my laboratory successfully, and participated in relevant projects in the CS.179J offering in Winter 2012. Three of the students have applied to the Ph.D. program in Computer Science and UCR. I have offered to be their faculty advisor, and I am in the process of trying to obtain grants to fund their Ph.D. studies over the next 5 years.*

*CS.179J; In the Spring 2010 quarter, the three students who worked on the project relating to programmable microfluidics identified several problems relating to resource-constrained scheduling. They were able to produce relatively naive solutions, which I was able to later improve upon with one of my graduate students. The work with my graduate student was later accepted for publication at DAC 2012. The undergraduate project was quite successful for a short 10-week capstone senior design.*

*In the Winter 2012 offering, both projects were even more successful. In one project, my graduate student developed a microfluidic synthesis framework which consists of three key steps to solve: scheduling, placement, and routing. The system was modular, so that algorithms could be developed and plugged in without affecting the rest of the system. All of the students who worked on this project were given papers describing existing algorithms and were asked to implement them. One student implemented two genetic algorithms used for scheduling. A second student implemented a simulated annealing-based placer. The third and fourth students implemented two routing algorithms each, one simple, and one complicated. The project was quite successful, and we hope to release the framework as open source for the community to use during the Summer of 2012. The student who worked on scheduling has applied to the Ph.D. program at UCR, as discussed above.*

*The second project developed algorithms to concurrently transport fluids in programmable microfluidic devices. With my help, the students modeled the problem as a vertex-disjoint path problem from graph theory, which is known to be NP-complete. The students developed two heuristics for this problem, which were inspired by the well-known PathFinder algorithm from FPGA routing. The results were quite encouraging, as this is the first significant effort to look at the problem of fluid routing in this particular context. I hope to submit a paper for publication to ICCAD 2012 based on this work, with the four students as co-authors. Three of the students who participated in this project have applied to the Ph.D. program at UCR. One has been working on traffic routing algorithms with another professor at CE-CERT, but has chosen to take an industrial position at ESRI instead; I hope to supervise the other two, as mentioned above.*

*\_ \_ \_ \_*

*RATIONALE: The founder of StopTheHacker.com approached me about my CS.179J course offered in Spring 2010 and suggested that I create new projects based on smart phone programming* (Self study authors note: The founder of StopTheHacker.com is faculty member Dr. Faloutsos)*. He offered to provide the API developed by his company so that it could be ported to smart phones.*

*IMPROVEMENT:  Initially, enrollment was low for the quarter, and there was some concern that the class would be cancelled. During the first week of class, I sent out an email announcement to the undergraduate student population to promote the class, and mentioned that smart phone programming would be supported.*

*RESULTS: Almost immediately, the size of the class doubled, and some students mentioned that they switched out of another CS 179J class to take this one because of the smart phone programming component.*

*Altogether, there were 4 smart phone programming projects, with teams comprised of 3 or 4 students. One of the projects, which was quite successful, ported the StoptheHacker.com API to Android phones, and the results were disseminated to representatives of the company.*

*Based on student feedback, there was significant satisfaction with the smart phone programming projects. I have encouraged my colleagues who teach other CS.179 courses (e.g. software systems, networking) to incorporate smart phone applications into their projects as well.*

*Two students who worked on a smart phone project, Richard Ramos and Jacob Leung, stayed at UCR to earn M.S. degrees; I supervised their M.S. project, which was also based on a smart phone; they implemented a tire pressure monitoring system prototype that notified the car owner, via a smart phone application, when the tire pressure moves out of the manufacturer-recommended range. I believe that this project grew out of their experience with smart phones in CS.179J.*

*\_ \_ \_ \_*

*RATIONALE: I received 10 Enforce Atom(™) Development boards from Intel as an equipment donation for use in my CS 179J offering for Spring 2012. At the same time, many students expressed interest in doing projects relating to computer vision using the MS Kinect's 3D camera system and API.*

*IMPROVEMENT: I created several projects around the development boards, including topics relating to computer vision based on the MS Kinect, and controlling a microfluidic device using a development board.*

*RESULTS:  The microfluidics project was successful; it ran on the development board seamlessly.*

*The Kinect-oriented projects were not successful. The development boards shipped with an obscure Linux variant (Timesys) that did not provide sufficient driver support. None of the groups working on Kinect-related projects were able to get the Kinect working with the development boards. One group tried a more traditional webcam and also failed. In response, I changed the projects around so that they could use desktop/laptop PCs running Windows instead, and all projects were successfully completed (although a significant amount of time was wasted trying to get the development boards to work).*

*A secondary problem was that the development boards became dangerously hot, and tended to overheat.*

*I do not plan to use the development boards again in my classes. I am planning to work with Harsha Madhyastha to integrate them into his cluster to provide a heterogeneous computing platform consisting of Intel Nehalem(™) and Atom(™) processors. This can be used to support research and undergraduate curriculum on operating system support for heterogeneous computing platforms.*

*If that project fails, I will ask an undergraduate student to assemble the development boards into a RAID array (each board has a 320GB hard drive), and I will use it as a data store/backup system for my laboratory.*

*I will definitely incorporate Kinect and camera-based projects into my future CS 179J courses; to keep the students' lives simple, I will require that they use Windows, because it is the only platform that has a mainstream and well supported Kinect SDK at the moment; any other operating system is likely to lead to more problems with device drivers, which can be a major time waste during a senior capstone design project class.*

End direct quotation of Dr. Brisk’s notes.

**Summary for this Section (*Section 5.C Continuous Improvement*):**

Our department has a culture of *data-driven* continuous improvement of undergraduate education. Faculty are rewarded for their efforts in improving the curriculum and in general teaching excellence, by having this aspect of their work be considered a significant factor in promotion and tenure decisions.

1. **Additional Information**

The ABET binders (described in detail in Section 4.B) for the last four years will be available for inspection. As noted above and shown in Figure 20, these binders are color coded, labeled on the spine and indexed in a consistent format, allowing rapid inspection. Note that these binders include sample student work (examples of graded high/medium/low quality of written submissions) for all courses.

|  |
| --- |
| IMG_0485[1] |

Figure 20: (a subset of) The ABET course reports and copies of all textbooks used in the CE program (*lower right*)

We will have examples of senior design projects available. A copy of every compulsory or recommend textbook used in the last four years will be available, and finally, typed minutes from meetings in the last four years will also be available for inspection.

# CRITERION 5. CURRICULUM

#### How our Curriculum Aligns with our Program Educational Objectives

The PEOs are designed to fulfill three main requirements: enable our graduates to pursue higher degrees, be successful in their industrial positions, and be good citizens of the community. The courses are structured to meet all the requirements. As evident from the course plan (see Table 40, Table 41 and Table 42), our students get a broad training in various aspects of CS that provide the technical foundations for both higher studies and industrial positions requiring such technical skills. Such courses include the basic science and math courses, the core CS courses, programming expertise, technical electives covering all the areas of the department, and a comprehensive senior design project. They are also required to take language and writing courses that are essential for success. We also prepare our students to be mindful of their ethical responsibilities and the broader role of engineering in society. These are covered in a number of courses, but most thoroughly in the Senior Design Project.

Additional information about how our PEOs align with our curriculum can be derived from considering how our curriculum maps to the student outcomes (Table 39) and how, in turn our student outcomes map to our PEOs (Table 14).

#### How our Curriculum Supports the Attainment of our Student Outcomes

As shown in Table 27, the required courses in the CS curriculum support attainment of all of the student outcomes. For each course, an **X** is entered in columns corresponding to the student outcomes for which that course supports attainment. Equivalently, for each student outcome, an **X** is entered in rows corresponding to the courses which support attainment of that outcome. It can be seen that attainment of each student outcome is supported by *at least* two courses.

Table 27: Curriculum Structure Supports Student Outcomes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | I | II | III | IV | V | VI | VII | VIII | IX | X | XI | XII | XIII | | **Lower** |  |  |  |  |  |  |  |  |  |  |  |  |  | | CS010 | X |  | X |  | X |  |  |  | X | X | X | X |  | | CS012 | X |  | X |  |  |  |  |  | X | X | X | X |  | | CS014 | X |  | X |  | X |  |  |  | X | X | X | X |  | | CS061 | X | X | X |  |  |  |  |  | X | X | X |  |  | | CS011 | X |  |  |  |  |  |  |  | X | X |  |  |  | | **Upper** |  |  |  |  |  |  |  |  |  |  |  |  |  | | ENGR 180 | X | X | X |  | X |  | X | X | X |  |  | X |  | | CS 100 | X |  |  |  | X |  | X |  | X | X |  |  | X | | CS 111 | X |  |  |  |  |  |  |  | X | X |  |  |  | | EE/CS 120B | X | X | X |  | X | X |  |  | X | X | X |  | X | | EE/CS 120A | X |  | X | X | X | X |  |  | X | X | X |  | X | | CS 122A | X | X | X |  | X | X |  |  | X | X | X | X | X | | CS 122B | X | X | X |  | X | X |  |  | X | X | X | X | X | | CS 130 | X | X | X |  |  |  |  |  | X | X | X |  |  | | CS 133 | X | X | X |  |  |  |  |  | X | X | X |  |  | | CS 141 | X | X | X | X |  |  |  |  | X | X | X | X |  | | CS 150 | X |  | X | X |  |  |  |  | X | X | X |  | X | | CS 152 | X | X | X |  |  |  |  |  | X | X | X |  |  | | CS 153 | X | X | X |  |  | X |  |  | X | X | X | X |  | | CS 160 | X | X | X |  |  | X |  |  | X | X | X | X |  | | CS 161 | X | X | X |  |  |  |  |  | X | X | X |  |  | | CS 161L | X | X | X |  |  |  |  |  | X | X | X | X |  | | CS 162 | X |  | X |  |  |  |  |  | X | X | X | X |  | | CS 164 | X |  |  |  |  |  |  |  | X | X |  | X |  | | CS 165 | X |  | X |  | X |  |  |  | X | X | X | X | X | | CS 166 | X | X | X |  |  |  |  |  | X | X | X |  |  | | CS 168 | X |  | X |  |  |  |  |  | X | X | X | X |  | | CS 170 | X | X | X |  |  |  |  |  | X | X | X | X |  | | CS 177 | X | X | X |  |  |  |  |  | X | X | X | X |  | | CS 179 | X | X | X | X | X | X | X | X | X | X | X | X | X | |

Note that we do *not* limit our evaluation of student outcomes to just these required classes. We evaluate the achievement of student outcomes in *every* technical elective that a CS student can take. See Appendix A for more details.

Table 28 shows the prerequisite structure of the program’s required lower division courses.

Table 28: Prerequisite Structure of Lower Division CS Required Courses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Course** | **Prereq 1** | **Prereq 2** | **Prereq 3** | **Prereq 4** | **Notes** |
| one of | ENGR 101-I | freshman standing |  |  |  |  |
|  | CS 100 | CS 014 |  |  |  |  |
|  | CS 141 | CS 014 |  |  |  |  |
|  | CS 150 | CS 014 | CS 111 | MATH 009C or | MATH 009CH |  |
|  | CS 152 | EE 100A |  |  |  |  |
|  | CS 153 | CS 061 | CS 100 | CS 111 |  |  |
| taken | CS 161 | EE/CS 120B |  |  |  |  |
| together | CS 161L |  |  |  |  |  |
|  | CS 111 | CS 010 | CS 011 | MATH 009C |  |  |
|  | EE/CS 120B | EE/CS 120A |  |  |  |  |
|  | EE/CS 120A | CS 061 |  |  |  | C or better |
|  | MATH 113 | MATH 010A |  |  |  | C or better |
|  | STAT 155 | MATH 009C or | MATH 009CH |  |  | may be concurrent |
| two of | MATH 046 | MATH 009B or | MATH 009BH |  |  | C or better |
|  | MATH 120 | MATH 010A | MATH 113 or | MATH 131 |  | C or better |
|  | MATH 126 | MATH/CS 011 |  |  |  | C or better |
|  | PHIL 124 | MATH/CS 011 or | CS/EE 120A or | PHIL 008 or | PHIL 008H | or instructors consent |
|  | PHYS 040B | PHYS 040A |  |  |  |  |
|  | PHYS 040A | MATH 008B, or | MATH009A or | MATH 09HA |  | may be concurrent |

Table 29 shows the prerequisite structure of the program’s required upper division courses.

Table 29: Prerequisite Structure of Upper Division CS Required Courses

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Course** | **Prereq 1** | **Prereq 2** | **Prereq 3** | **Prereq 4** | **Notes** |
| CS 100 | CS 014 |  |  |  |  |
| CS 122A | CS 012, or | CS 013 | EE/CS 120B |  | C or better |
| CS 122B | CS 122A |  |  |  |  |
| CS 130 | CS 100 | MATH 113 |  |  | or instructor's consent |
| CS 133 | CS 100 | CS 111 | MATH 113 |  |  |
| CS 134 | CS 130 |  |  |  |  |
| CS 145 | CS 141 | MATH 113 or | MATH 131 |  |  |
| CS 160 | CS 061 | CS 100 | CS 111 |  |  |
| CS 162 | CS 161 | CS 161L |  |  | C- or better |
| CS 164 | CS 100 | CS 111 | CS 153 |  |  |
| CS 165 | CS 141 | CS 153 |  |  |  |
| CS 166 | CS 100 | CS 111 |  |  |  |
| CS 168 | EE/CS 120A |  |  |  | or instructor's consent |
| CS 169 | CS 164 |  |  |  | or instructor's consent |
| CS 170 | CS 100 | CS 111 |  |  |  |
| CS 177 | CS 100 | CS 111 |  |  | C++ prog' proficiency |
| CS 179E | CS 141 | CS 152 | ENGR 180W |  | C- or better; |
| CS 179F | CS 153 | ENGR 180W |  |  | 8 additional |
| CS 179G | CS 141 | CS 166 | ENGR 180W |  | upper-division |
| CS 179I | CS 141 | CS 164 | ENGR 180W |  | units in CS; |
| CS 179J | CS 100 | CS 111 | CS 122A | CS 161 | CS 160 recommended |
| CS 179K | CS 180 | ENGR 180W |  |  |  |
| CS 179M | CS 100 | CS 111 | CS 170 | ENGR 180W |  |
| CS 179N | CS 130 | ENGR 180W |  |  |  |
| CS 180 | CS 014 | CS 100 |  |  |  |
| CS 181 | CS 061 | CS 100 | CS 111 | CS 150 |  |
| CS 183 | CS 100 |  |  |  |  |
| CS 193 | CS 141 | consent of instructor |  |  |  |
| EE 140 | senior standing in CE, CS, or EE |  |  |  |  |
| MATH 120 | MATH 010A | MATH 113 or | MATH 131 |  | C- or better; |
| MATH 135A | CS 010 | MATH 113 or | MATH 131 |  | MATH 113/131 may |
| MATH 135B | CS 010 | MATH 113 or | MATH 131 | MATH 135A | be taken concurrently |

Table 30 shows the suggested course plan for a CS major.

Table 30: The Suggested Course Plan for a CS Major

|  |
| --- |
|  |

**How our program meets the requirements in terms of hours and depth of study for each subject area**

**General Education.** UCR has defined a set of General Education requirements that all students at UCR must satisfy as part of their baccalaureate (Bachelor's) degree program. This set of courses in the General Education consists of 37 credit hours, and is designed to provide all undergraduates with a balanced, broad and coherent general education, and to promote competence in reading, writing, speaking, listening, critical thinking, computer literacy and mathematics.

The core courses are intended to provide a broad base so that those who terminate their formal education with the Bachelor's degree can continue to grow as professionals throughout their careers. This goal is met by a curriculum in which there is a progression of coursework in which fundamental knowledge acquired in earlier years is applied in later courses. Satisfaction of these goals also implies success in attaining the objectives of the program.

For concreteness we address the Program Criteria for Computer Science (and Similarly Named Computing Programs) one-by-one below. For clarity we have formatted the ABET text in a different font and color.

Student Outcomes

The program must enable students to attain, by the time of graduation:

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

Recall that this outcome (j) is our student outcome X, thus we measure this outcome extensively. For example, see Table 31, Table 32, Table 34, Table 35, Table 36 and Table 37. As we can see in Table 39, this outcome is measured in virtually every course offering in the computer science program.

Concretely, the relevant skills are taught in the following *required* courses.

* mathematical foundations: MATH113, CS111, STAT155, MATH009A/ 009B/ 009C/ 010A
* algorithmic principles: CS010, CS012, CS014, CS141
* computer science theory: CS141, CS150
* design of computer-based systems: CS061, CS153, EE/CS120A, EE/CS120B, CS161/161L
* comprehension of the tradeoffs involved in design choices: CS061, CS161/161L, EE/CS120A, EE/CS120A, CS 153.

Finally, we note that CS179 was design explicitly to cover (and test) *all* these skills.

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

Recall that this outcome (k) is our student outcome XI, thus we measure this outcome extensively. For example, see Table 30, Table 34, Table 35. As we can see in Table 27, this outcome is measured in a majority of course offering in the computer science program.

Concretely this skill is taught in the following *required* courses.

* CS100: *Software Construction*: in which students learn design, coding layout, and style; implementation strategies; quality attributes; prototyping, reuse, and components; debugging, testing, and performance; integration and maintenance; documentation; standards, analysis, and selection of tools and environment; and personal software processes.
* CS153: *Design of Operating Systems*: in which students learn principles underlying the design of operating systems with a focus on principles and mechanisms used throughout the design process.
* CS152: *Compiler Design*: in fundamentals of compiler design, including lexical analysis, parsing, semantic analysis, compile-time memory organization, run-time memory organization, code generation, and compiler portability issues.
* CS 161/CS 161L: *Design and Architecture of Computer Systems*: in which students learn operation and design of instruction set architectures, CPUs, cache memories, main memories and disks. At every step they are exposed to the cost performance tradeoffs involved in the design choices. In the laboratory course (CS 161L) has the students building and evaluating these systems.

Finally, we note that CS179 was design explicitly to cover (and test) this skill.

**Curriculum**

Students must have the following amounts of course work or equivalent educational experience:

a. Computer science: One and one-third years that must include:

1. Coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture.

We cover the relevant topics in the following *required* courses:

* fundamentals of algorithms: CS010, CS012, CS014, CS141, CS150, CS152
* data structures: CS014, CS141
* software design: CS100
* concepts of programming languages CS010, CS012, CS100, CS150, CS 152
* computer organization and architecture CS061, EE/CS120A, EE/CS120B, CS161/161L

2. An exposure to a variety of programming languages and systems.

Considering only the following *required* courses, our students are exposed to:

* C: CS012, CS120B
* C++: CS010, CS014, CS061, CS100, CS122A, CS153, CS161/161L
* Python: CS141 (for the last two quarters and going forward, before that we used C++ )
* VHDL: CS122A
* LC-3 (assembly language): CS061, EE/CS 102B

Thus our students have an exposure to a wide variety of programming languages and systems.

3. Proficiency in at least one higher-level language.

Considering only the following *required* courses, our graduates will have spent 34 units of classes that use C++.

* C++ is used this following required classes: CS010, CS012, CS014, CS061, CS122A, CS141, CS153 and CS161/161L

Thus at a minimum, our graduates have had experience programming in C++ for operating system, embedded systems and general programming.

#### 4. Advanced course work that builds on the fundamental course work to provide depth.

#### b. One year of science and mathematics:

#### 1. Mathematics: At least one half year that must include discrete mathematics. The additional mathematics might consist of courses in areas such as calculus, linear algebra, numerical methods, probability, statistics, number theory, geometry, or symbolic logic. [CS]

As shown in Table 28 and Table 30, in order to graduate CS students must have at least MATH 011 (*Intro to Discrete Structures*), MATH 113 (*Linear Algebra*), CS111 (Discrete Structures). They must also have STAT 155 (*Probability and Statistics for Science and Engineering*).

Moreover our graduates must have at least *two* of:

* MATH 046 (*Introduction to Ordinary Differential Equations*)
* MATH 120 (*Optimization*)
* MATH 126 (*Introduction to Combinatorics*)
* PHIL 124 (*Formal Logic*)

Because of the prerequisites of the above (as show in Table 28) and College-level requirements, our students must also take one full year of Calculus, including MATH 009A, MATH 009B, MATH 009C, and MATH 10A (*Calculus of Several Variables*).

It may be helpful to review this informal with mappings to the ABET text.

* Discrete mathematics: Math/CS 11, Math/CS 111
* Calculus: Math9A-C, Math10A
* Linear algebra: Math 10A, Math 113
* Probability/statistics: Math/CS11, STAT 155
* Number theory: Math/CS11, Math/CS 111
* Symbolic logic: Math/CS11

Thus, the absolute *minimum* number of units in mathematics a computer science student needs to graduate is 40, significantly more than the ABET requirement.

#### 2. Science: A science component that develops an understanding of the scientific method and provides students with an opportunity to experience this mode of inquiry in courses for science or engineering majors that provide some exposure to laboratory work.

#### Considering only *required* courses:

#### The required sequence of three courses physics in physics, PHYS 040A (Mechanics), PHYS 040B (Heat/Waves/Sound), PHYS 040C (Electricity/Magnetism) all have a significant hands-on laboratory section illustrating experimental foundations of physical principles and selected applications. In total, this requires 90 hours of laboratory work.

#### Moreover while CS161 is a lecture class, student must also take CS161L at that same time, this requires a total of 30 hours of laboratory work. Likewise EE/CS120A and EE/CS 120B *each* require 60 hours of laboratory work, involving use of hardware description languages, synthesis tools, programmable logic, and significant hardware prototyping. Finally CS179 requires a minimum of 120 hours of lab work.

#### Thus, the minimum number of hours of laboratory experience required by one of our graduates is 360 hours. In practice, depending on the technical electives chosen (see Table 30) graduates typically have more than 580 hours of laboratory experience.

#### Note that we have *not* counted “computer labs” in the above. By “computer lab” we mean a course obligation that “only” involves computer programming.

#### 

**Design Experience: CS 179**:

The CS Senior Design Project (CS179 E-Z) is a one-quarter course that provides students the experience of designing a real-life project.

The catalog description of each CS179 offering begins with the following overarching vision for the offering:

*Under the direction of a faculty member, student teams propose, design, build, test, and document software and/or hardware devices or systems. Emphasizes professional and ethical responsibilities; the need to stay current on technology; and its global impact on economics, society, and the environment.*

The project has the following options for concentrations:

* **CS 179E. Compilers**: Covers the planning, design, implementation, testing, and documentation of a compiler-related system.
* **CS 179F. Operating Systems**: Covers the planning, design, implementation, testing, and documentation of an operating-related system.
* **CS 179G. Database Systems**: Covers the planning, design, implementation, testing, and documentation of a database-related system.
* **CS 179I. Networks**: Covers the planning, design, implementation, testing, and documentation of a network-related system.
* **CS 179J. Computer Architecture and Embedded Systems**: Covers the planning, design, implementation, testing, and documentation of a computer architecture and embedded systems-related system
* **CS 179K. Software Engineering**: Covers the planning, design, implementation, testing, and documentation of a software engineering related system
* **CS 179M. Artificial Intelligence:** Covers the planning, design, implementation, testing, and documentation of an artificial intelligence related system.
* **CS 179N. Graphics and Electronic Games**: Covers the planning, design, implementation, testing, and documentation of a graphics- or electronic game-related system

Each concentration has different prerequisites, ensuring the students have the requisite background classes taken. In addition, senior standing in Computer Science is required.

Projects are either suggested by the instructors or proposed by students and approved by the instructors. The instructors ensure that all design projects have sufficient level of technical difficulty and make use of knowledge and skills from earlier computer science courses.

**Objectives**

The Senior Design Project is the culmination of coursework in the bachelor’s degree program in computer science. Students are expected to apply the concepts and theories of computer science to a real-world design project. Detailed written reports, pseudocode, prototypes, test reports, working demonstration, and oral presentations are required.

The following are the specific course objectives.

1: Balancing design tradeoffs: cost performance schedule and risk

2: Writing project proposals

3: Team-project organization and management (including time lines)

4: Requirements capture and analysis

5: Design and architecture

6: Prototyping (possibly via simulation)

7: Verification/validation

8: Writing and presenting final reports

9: Engineering professionalism and responsibility

10: Engineering careers and the modern world

Table 31 below shows the mapping between the course objectives and the Student Outcomes. The value ‘3’ indicates a strong mapping, ‘2’ and ‘1’ a weaker mapping (zero mappings are left blank.

Table 31: The mapping between the CS 179 course objectives and the Student Outcomes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Outcome related learning objectives | I | II | III | IV | V | VI | VII | VIII | IX | X | XI | XII | XIII | | 1: balancing design tradeoffs: cost performance schedule and risk | 2 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 0 | 0 | | 2: writing project proposals | 0 | 3 | 0 | 3 | 0 | 3 | 0 | 0 | 3 | 3 | 3 | 0 | 0 | | 3: team-project organization and management (including time lines) | 0 | 0 | 2 | 3 | 0 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | | 4: requirements capture and analysis | 2 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 0 | 0 | | 5: design and architecture | 2 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 0 | 0 | | 6: prototyping (possibly via simulation) | 3 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 0 | | 7: verification/validation | 3 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 0 | | 8: writing and presenting final reports | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | | 9: engineering professionalism and responsibility | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 1 | | 10: engineering careers and the modern world | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 3 | 3 | 0 | 3 | 0 | 3 | |

**Credits and Hours**

Four quarter units of credit are granted for the completed project and other required components listed here. It is expected that approximately twelve hours of laboratory work will be required weekly for satisfactory completion of the project.

**Weekly Class Meetings**

The entire class of CS179 meets once each week for one hour. These meetings are intended to provide instruction in topics common to all design projects (engineering economics, ethics, etc.). In addition, it is required that each project team meet with their faculty supervisor on a weekly basis to report and discuss the progress of the project. They may include brief presentations by each team, aimed at improving technical presentation skills. Attendance of the lectures and weekly meetings is mandatory.

**Project Participants**

Projects are completed in small teams with shared responsibility. If the team option is elected, each student will be held responsible for a distinct component of the total team effort. Team projects will be sufficiently more complex than individual projects so as to allow for an appropriate workload for all team members.

**Project Elements**

The senior design projects include proposal and report writing, experiment design, (hardware, were appropriate) and software design, test plan and test, broad impact and ethical issues, among other things. Recall that this is a design course and students must define a ***design*** project, not a research, nor an evaluation or fabrication project. It is designed to be a balanced approach to encompass all of the elements stated above.

Each design project must include the following components:

1. **A Clear Technical Design Objective and the Project Contract** (Contract due on Monday of week 2): Each group must identify a design project and sign the Contract by the due date, and should have good estimated answers to the following questions and obtain the endorsement of the section professor:

* Is the objective achievable within the time frame?
* Does the group have the expertise to complete the design, prototype, and testing?
* Does the group have access to the financing for the prototype? (where appropriate)
* Does the group have access to the required test equipment? (where appropriate)
* Is this a *design* problem (not research, nor fabrication)?
* Is the project significant enough to be worthy of four credits (12 hours/week/person)?

1. **Experiment Design and Feasibility Study** (Required section in Final Report, 5% of final grade) Design and carry out experiments to evaluate the feasibility of project ideas, alternatives, trade-offs and realistic design constraints. Analyze the experimental results to prove the feasibility of your project idea and select the best solution to be further developed in the design project.
2. **A Detailed Design Specification** (Due in week 3): Describes the functions and quantitatively measurable design objectives, design methods, hardware and software architecture and interfaces, user interface, realistic constraints in terms of time, cost, safety, reliability, social impact, ethics, etc. It must also list and consider the industry standards related to your project, including hardware, protocols, software and tools (e.g., 802.11, RS232, USB, PCI, 3G, API, device drivers, VHDL).
3. **Global, Economic, Environmental and Societal Impact** (Due on Monday of week 4, 2% of final grade):*Each* student must write an essay (500 or more words) providing an analysis of the potential global, economic, societal, and environmental impact of the project. You do not need to address every aspect, just focus on several aspects that are related to your project. For example, if your project is made into a product, how will it improve quality of life, affect the environment, enhance entertainment, education, globalization etc? Are there any ethical or political debates, laws and regulations that are related to your project?
4. **Contemporary Engineering Issues** (Due on Monday of week 4 of the winter quarter, 2% of final grade)Write an essay (500 or more words) on the contemporary engineering issues related to the project. Potential contemporary engineering issues related to your project are new technologies, new industry standards, new design methods, new materials, new trends in manufacturing, etc.
5. **Test Plan** (Required section in Final Report, 5% of final grade)**:** A detailed description of your design of experiments to test and measure whether the final product and each of its components meet the design specifications, and, if not, to test and measure the errors and deviations from specifications.
6. **Understanding of Professional and Ethical Responsibility** (Required section in Final Report, see grading below)Write an essay (500 or more words) on (a) what are the ethical implications of your project, (b) how you addressed them, and (c) what you learned through this design project about professional and ethical responsibility.
7. **Recognition of the need for and an ability to engage in lifelong learning** (Required section in Final Report, 2% of final grade)Write an essay (200 or more words) on how doing this design project helped you (a) recognize the need and (b) developed the ability in lifelong learning.
8. **Design Review Presentation** (Week 9 or 10, 5% of final grade): Each group must make a presentation of its design specification and progress to faculty and other students. Requirements of design review presentation will be provided.
9. **Detailed Quantitative Design and Prototype** (To be completed before week 7 at the latest): Each component of the selected solution and the overall system should be designed and implemented. In most cases, it is necessary to construct a system prototype (or component prototype).
10. **Test Report** (Due week 9, 5% of final grade):Carry out the Test Plan you developed to identify how well your final design meets the specifications under the defined constraints, and present the results in this report.
11. **Final Presentation** (Week 10, 5% of final grade): Each group must make a presentation of the final design and show a working demo to faculty and other students. Requirements of final presentation will be provided.
12. **Working Demo and Final Report** (Due on Wednesday of the finals week before 5pm,): The final report must include all the required sections and appendices in a template file, final presentation ppt file and video or data of a working demo must be archived into the project binder. A working demo of the completed design is critical, it is convincing evidence that your design is completed and works. The demo should show whether and how design specifications are met.

**Grading**

In addition to the deliverables listed above, each project will also be graded on the following:

1. **Laboratory Notebook, Weekly Progress and Lecture Attendance:** Each student team needs to maintain a laboratory notebook for the duration of their projects and report progress to the section instructor at least weekly. Each week, you must show evidence of amount of work done and progress in the design, implementation and/or testing. Attendance of the lectures is mandatory. Everyone must sign in at each lecture. (This portion accounts for 7.5% of grade).
2. **Professional Ethics and Responsibility** (7.5% of the final grade): Each student will be evaluated by their fellow team member(s) and by your section instructor. An evaluation form that explains how this is graded is handed out on the first week

Grading is determined by the instructor for each project and student. Note that grades are assigned to an individual, not to a project.

**CS179 Project Topics**

Projects topics depend on the concentration. Typical examples include:

**CS 179G. Database Systems**: Design and build a spatial database that would allow the owner of a vending machine company to manage an inventory of 30,000 machines, serviced by 20 technicians.

**CS 179M. Artificial Intelligence:** Design and build an intelligent tutoring system to teach non-native speakers sentence tagging.

**CS 179M. Artificial Intelligence:** Design and build a path-finding algorithm that optimizes the loading and unloading of container ships.

**CS 179J. Computer Architecture and Embedded Systems**: Fluid routing algorithms for programmable microvalve arrays.

**CS 179J. Computer Architecture and Embedded Systems**: Create electronic blocks that assist in at-home care, including detectors of various sounds (alarms, knocks, screams, etc.), out-of-bed detector, fall detector, etc. Blocks interface with an existing web-based system.

**CS 179J. Computer Architecture and Embedded Systems**: Human detection using webcam and/or MS Kinect.

**CS 179I. Networks**: Design and build a distributed peer to peer system that can collaboratively store, search for, and download a file.

**CS 179I. Networks**: Design and implementation of an application layer multicast protocol for information sharing..

**CS 179K. Software Engineering**: Write the specification, design, test plan, user manual, and code for a stand-alone system to manage bibliographical references to be processed by bibtex/latex.

**CS 179K. Software Engineering**: Write the specification, design, test plan, user manual, and code for a decision diagram library.

**CS 179N. Graphics and Electronic Games**: Teams design and implement a complete video game following practices of good software engineering and iterative user testing. They also interface with sound and art talents to realize their vision.

**Materials that will be available for review during the visit to demonstrate achievement related to this (Design Experience) criterion.**

For the CS179 option, we have annotated *physical* binders that will be placed in the ABET evaluators room. Theses binders contain the syllabus, graded homework and lab assignments with solutions, student outcomes, sample student work, assessment of student outcomes, and changes that have been made and are recommended for the future.

Table 5-1 Curriculum for Computer Science: Part I

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COMPUTER SCIENCE COURSES | | **R**equired **E**lective **S**elected **E**lective | Subject Area | | |  | Last Two Terms the Course was Offered: | Maximum Section Enrollment for Last Two Terms |
| Department and Course Number | Title |  | Math & Basic Sciences | comp Topics | Gen Ed | Other | |  |
| **FALL QUARTER, YEAR 1** | |  |  |  |  |  |  |  |
| CS 010 | C++ Programming | R |  | F |  |  | W2012, F2011 | 240, 242 |
| ENGL 001A | Beginning Composition | R |  |  | X |  | W2012, F2011 | 917, 1136 |
| ENGR 001I | Professional Development & Mentoring | R |  |  |  | X | F2011, F2010 | 85, 99 |
| MATH009A | First-Year Calculus | R | X |  |  |  | W2012, F2011 | 275, 440 |
| **WINTER QUARTER, YEAR 1** | |  |  |  |  |  |  |  |
| CS 012 | C++ Programming II | R |  | F |  |  | W2012, F2011 | 114, 27 |
| ENGL 001B | Intermediate Composition | R |  |  | X |  | W2012, S2011 | 1716, 2145 |
| MATH 009B | First-Year Calculus | R | X |  |  |  | W2012, F2011 | 848, 756 |
| MATH/CS 011 | Introduction to Discrete Structures | R | X |  |  |  | W2012, F2011 | 60, 71 |
| **SPRING QUARTER, YEAR 1** | |  |  |  |  |  |  |  |
| CS 014 | Introduction to Data Structures & Algorithms | R |  | F |  |  | W2012, F2011 | 29, 59 |
| MATH 009C | First-Year Calculus | R | X |  |  |  | W2012, F2011 | 407, 423 |
| Breadth \_\_\_\_\_\_\_\_\_\_ |  | R |  |  | X |  |  |  |
| **FALL QUARTER, YEAR 2** | |  |  |  |  |  |  |  |
| CS 061 | Machine Organization & Assembly Language Programming | R |  | A |  |  | W2012, F2011 | 88, 61 |
| CS 100 | Software Construction | R |  | X |  |  | W2012, F2011 | 30, 57 |
| PHYS 040A | General Physics | R | X |  |  |  | W2012, F2011 | 354, 283 |
| Breadth \_\_\_\_\_\_\_\_\_\_ |  | R |  |  | X |  |  |  |
| **WINTER QUARTER, YEAR 2** | |  |  |  |  |  |  |  |
| CS 111 | Discrete Structures | R |  | A |  |  | W2012, F2011 | 50, 42 |
| EE/CS 120A | Logic Design | R |  | A |  |  | W2012, F2011 | 60, 60 |
| PHYS 040B | General Physics | R | X |  |  |  | W2012, S2011 | 260, 345 |
| Breadth \_\_\_\_\_\_\_\_\_\_ |  | R |  |  | X |  | W2012, F2011 | 50, 42 |
| **SPRING QUARTER, YEAR 2** | |  |  |  |  |  |  |  |
| CS/EE 120B | Embedded Systems | R |  | A |  |  | W2012, S2011 | 67, 30 |
| PHYS 040C | General Physics | R | X |  |  |  | F2011, S2011 | 289, 244 |
| Breadth \_\_\_\_\_\_\_\_\_\_ |  | R |  |  | X |  |  |  |
| **FALL QUARTER, YEAR 3** | |  |  |  |  |  |  |  |
| CS 141 | Intermediate Data Structures & Algorithms | R |  | A |  |  | F2011, S2011 | 30, 31 |
| CS 161 & CS 161L | Design & Architecture of Computer Systems & Lab | R |  | A |  |  | F2011, S2011 | 29, 17 |
| MATH 010A | Multivariable Calculus | R | X |  |  |  | W2012, F2011 | 331, 257 |
| ENGR 101I | Professional Development & Mentoring | R |  |  |  | X | F2011, F2010 | 34, 33 |
| **WINTER QUARTER, YEAR 3** | |  |  |  |  |  |  |  |
| CS 150 | Theory of Automata & Formal Language | R |  | A |  |  | W2012, S2011 | 48, 28 |
| CS 153 OR CS 160 | Design of Operating Systems | R |  | A |  |  | W2012, S2011 | 29, 13 |
| Technical Elective\*\* |  | R |  |  |  |  |  |  |
| Math Elective\*\* |  | R | X |  |  |  |  |  |
| **SPRING QUARTER, YEAR 3** | |  |  |  |  |  |  |  |
| Engineering Elective |  | R |  |  |  |  |  |  |
| ENGR 180W | Technical Communications | R |  |  |  |  | F2011, W2011 | 48, 47 |
| MATH 113 | Linear Algebra | R | X |  |  |  | F2011, S2011 | 62, 48 |
| Technical Elective\*\* |  | R |  |  |  |  |  |  |
| **FALL QUARTR, YEAR 4** |  |  |  |  |  |  |  |  |
| STAT 155 | Probability & Statistics for Engineers | R | X |  |  |  | W2012, F2011 | 99, 94 |
| Technical Elective\*\* |  | R |  |  |  |  |  |  |
| Technical Elective\*\* |  | R |  |  |  |  |  |  |

**Table 5-1 Curriculum for Computer Science: Part II**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COMPUTER SCIENCE COURSES | | **R**equired **E**lective **S**elected **E**lective | Subject Area | | |  | Last Two Terms the Course was Offered: | Maximum Section Enrollment for Last Two Terms |
| Department and Course Number | Title |  | Math & Basic Sciences | Eng Topics | Gen Ed | Other | |  |
|  | |  |  |  |  |  |  |  |
| Breadth \_\_\_\_\_\_\_\_\_\_ | | R |  |  | X |  |  |  |
| **WINTER QUARTER, YEAR 4** | |  |  |  |  |  |  |  |
| CS 152 | Compiler Design | R |  |  |  |  | W2012, W2011 | 40, 30 |
| Math Elective\*\* |  | R | X |  |  |  |  |  |
| Technical Elective\*\* |  | R |  |  |  |  |  |  |
| Breadth \_\_\_\_\_\_\_\_\_\_ |  | R |  |  | X |  |  |  |
| **SPRING QUARTER, YEAR 4** | |  |  |  |  |  |  |  |
| CS 179 | Project in Computer Science | R |  | A |  |  |  |  |
| Technical Elective\*\* |  | R |  |  |  |  |  |  |
| Breadth \_\_\_\_\_\_\_\_\_\_ |  | R |  |  | X |  |  |  |

# 

**Table 5-1 Curriculum for Computer Science: Part III**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COMPUTER SCIENCE COURSES | | **R**equired **E**lective **S**elected **E**lective | Subject Area | | |  | Last Two Terms the Course was Offered: | | Maximum Section Enrollment for Last Two Terms |
| Department and Course Number | Title |  | Math & Basic Sciences | Com topics | Gen Ed | Other | | |  |
|  | |  |  |  |  |  | |  |  |
| **TECHNICAL ELECTIVES** | |  |  |  |  |  | |  |  |
| CS 122A | Intermediate Embedded & Real-Time Systems | SE |  | A |  |  | | F2011, F2010 | 30, 44 |
| CS 122B | Advanced Embedded & Real-Time Systems | SE |  | A |  |  | | W2007, W2006 | 10, 13 |
| CS 130 | Computer Graphics | SE |  | A |  |  | | W2011, F2009 | 30, 51 |
| CS 133 | Computational Geometry | SE |  | A |  |  | | W2009, W2008 | 6, 5 |
| CS 134 | Video Game Creation & Design | SE |  | A |  |  | | W2009, W2008 | 12, 23 |
| CS 145 | Combinatorial Optimization Algorithms | SE |  | A |  |  | | S2007 | 16 |
| CS 151 | Introduction to Theory of Computation | SE |  | A |  |  | |  |  |
| CS 160 | Concurrent Programming & Parallel Systems | SE |  | A |  |  | | W2011, S2009 | 13, 19 |
| CS 162 | Computer Architecture | SE |  | A |  |  | | S2006, S2004 | 10, 14 |
| CS 164 | Computer Networks | SE |  | A |  |  | | W2012, W2011 | 30, 31 |
| CS 165 | Computer Security | SE |  | A |  |  | | F2011, F2010 | 29, 21 |
| CS 166 | Database Management Systems | SE |  | A |  |  | | F2011, S2011 | 32, 50 |
| CS 168 | Introduction to VLSI Design | SE |  | A |  |  | | W2004, W2003 | 6, 19 |
| CS 169 | Mobile Wireless Networks | SE |  | A |  |  | | S2011 | 10 |
| CS 170 | Introduction to Artificial Intelligence | SE |  | A |  |  | | W2012, W2011 | 35, 10 |
| CS 177 | Modeling & Simulation | SE |  | A |  |  | | W2012, S2011 | 30, 21 |
| CS 179 E-Z | Project in Computer Science | SE |  | A |  |  | |  |  |
| CS 180 | Introduction to Software Engineering | SE |  | A |  |  | | W2012, W2011 | 31, 30 |
| CS 181 | Principles of Programming Languages | SE |  | A |  |  | | S2010, W2009 | 23, 7 |
| CS 183 | UNIX System Administration | SE |  | A |  |  | | F2011, F2010 | 34, 33 |
| CS 193 | Design Project | SE |  | A |  |  | | W2012, F2011 | 4, 5 |
| EE 140 | Computer Visualization | SE |  | A |  |  | | F2009, S2007 | 19, 3 |
| MATH 120 | Optimization | SE | X |  |  |  | | F2011, S2011 | 55, 74 |
| MATH 135A | Numerical Analysis | SE | X |  |  |  | | F2011, F2010 | 31, 20 |
| MATH 135B | Numerical Analysis | SE | X |  |  |  | | W2012, W2011 | 5, 4 |
| OVERALL CREDIT HOURS FOR COMPLETION OF THE PROGRAM | | 182 |  |  |  |  | |  |  |
| PERCENT OF TOTAL |  |  | 26 | 37 | 20 | 17 | |  |  |

# CRITERION 6. FACULTY

1. **Faculty Qualifications**

The Computer Science Program possesses a wealth of talent, experience, and education in its faculty. The faculty has a wide diversity of emphasis areas within computer science fields. This allows us to cover all major areas in our curriculum with ease. In fact, for *every* course offering in CS, there are *at least* three faculty that can teach it.

The Department of Computer Science currently has five lecturers and twenty-two tenure-track faculty that include three ACM Fellows, four IEEE Fellows, five AAAS Fellows, one NSF Presidential Young Investigator, one AFOSR Young Investigator, and seven NSF CAREER award holders.

While many of our faculty are famous for contributions to research. Their achievements in teaching are no less impressive. For example, Dr. Vahid’s book, *Embedded Systems Design* (Vahid/Givargis, Wiley) is used in more than 50 universities, and his textbook *Digital Design* (Vahid, Wiley) is has been adopted in more than 60 universities worldwide. Similarly Dr. Keogh’s teaching materials on *Introductory Artificial Intelligence* have been used in at least 60 universities worldwide, and a quick web search reveals Spanish, Portuguese, Chinese and Polish translations have been created by others.

***6.A.1 An Overview of the Personnel Changes made since the last ABET Visit***

Since the last ABET visit, the Computer Science Department lost:

* Dimitrios Gunopulos: *Databases*.
* Vana Kalogeraki: *Operating Systems*/ *Distributed Systems*.
* Brett Fleisch: *Computer* and *Network Systems*.
* Teodor Przymusinski: *Artificial Intelligence*.

Dr. Gunopulos and Dr. Kalogeraki (a married couple) left in 2010 to return to Greece to start a family. Dr. Fleisch left in 2011 to pursue an opportunity in Thailand. Dr Przymusinski retired in 2010. While he remains an emeritus professor, he is not active and does not retain an office at UCR.

We also had:

* Thomas Payne: *Programming Languages*.

retire and become an Emeritus Professor in 2010. However, Dr. Payne retains an office and remains *very* active, teaching one class per quarter, attending all faculty meetings and doing other department service. His retirement is just a bookkeeping device based on UC retirement rules.

Since the last ABET cycle we hired:

* Iulian Neamtiu: *Software Engineering*.
* Philip Brisk: *Embedded Systems/Computer Architecture*.
* Vagelis Hristidis: *Databases*/*Information Retrieval*
* Harsha Madhyastha: *Distributed Systems/Networking/Security*.
* Tamar Shinar: *Scientific Computing*/*Computer Graphics*.

Below we briefly discuss why these faculty were hired, and the contributions to our strengths in undergrad teaching. Their significant contributions to undergraduate research are discussed elsewhere in this document.

**Dr. Iulian Neamtiu** was hired in 2008 to strengthen the existing Software Engineering group. Dr. Neamtiu has adjusted the lecture material for CS 180 to increase the use of formal methods in software construction. The use of formal methods in software specification and software validation leads to a better understanding of software requirements and software behavior, and prepares students to write higher quality software, i.e., more usable and less prone to errors. Moreover, the emphasis on rigor and formality gives students an opportunity to use their training and knowledge in mathematics and logic towards solid software construction methods. For the project part of the course, Dr. Neamtiu has chosen an approach that emphasizes flexibility in team formation and implementation strategy, while adhering to strict documentation and schedule guidelines; this approach includes the role of exposing students to realistic software development practices, as well as issues and solutions that appear in the development of large projects in a multi-person team. Projects change each quarter and topics are highly relevant to what students will work on after graduation. For example, recent team projects included implementing “lite" versions of LinkedIn, Facebook, YouTube, Twitter, Blockbuster online store, Netflix, IMDB. Other projects involved developing Android and iPhone applications for augmented reality, time management, and restaurant reservations. The approach has already started to bear fruit, with outgoing students indicating they were able to find jobs and internships based on the large, team-oriented, highly-relevant project topics in CS 180.

**Dr. Philip Brisk** was hired in 2009 to strengthen the existing embedded systems, architecture, and CAD group. Dr. Brisk's primary undergraduate teaching responsibility has been CS.120B; he has integrated learning materials developed by his colleague, Dr. Vahid, into the course. Through NSF funding Dr. Vahid and Dr. Tony Givargis (UC Irvine) developed an electronic textbook on undergraduate embedded systems which focuses on the usage of state machines as design patterns for microcontroller programming; the textbook also introduces fundamental concepts such as fixed-point arithmetic (in software), real-time systems, concurrent state machines, digital signal processing, and control theory. Dr. Vahid and Dr. Givargis also developed a software learning suite comprised of the Riverside-Irvine Builder of State Machines (RIBS) and the Riverside-Irvine Microcontroller Simulator (RIMS), which are used extensively in CS.120B. Although the development of these learning materials pre-dates Dr. Brisk's appointment at UCR, he has promoted their use and adoption in CS.120B. Students use RIBS and RIMS for homework assignment, and to assist them in the laboratory sections where they program state machine applications using AVR microcontrollers.

Dr. Brisk also teaches the CS 179J senior design project course on computer architecture and embedded systems. Dr. Brisk has introduced several new project options into CS 179J, included the smart phone application development for the Android platform, which was received quite favorably by the students. Dr. Brisk has also developed projects around his research on programmable microfluidics, which expose undergraduates to an exciting and emerging interdisciplinary technology. Finally, Dr. Brisk has introduced video processing projects as options for CS 179J, which include the use of webcams and Microsoft Kinect 3D cameras. Dr. Brisk obtained an equipment donation from Intel for use in present and future CS 179J offerings. The equipment includes 10 Intel Atom processor development boards, and one Intel Atom E600 "Stellarton" development, which includes an lntel Atom processor and Altera Arria FPGA integrated into the same package. Dr. Brisk has used this equipment in video processing projects in the Winter 2012 offering of CS 179J, and they will be made available for future offerings of CS 179J as well.

**Dr. Vagelis Hristidis** was hired in 2011 partly to replace the loss of Dr. Gunopulos, but also to give us more of a presence in information retrieval. Since coming to UCR Dr. Hristidis created new course CS 172, “Introduction to Information Retrieval,” which was offered for the first time in Spring 2012. This course teaches students the cutting edge technologies employed in Web search engines and other domain-specific search systems. In addition to examinations, this course involves group projects, where students study and develop various aspects of Information Retrieval. Further, Dr. Hristidis has experience in teaching Database Systems and Data Structures courses. He has also been involved in undergraduate research activities funded by the NSF, as supplements to his research grants. In the past, he has mentored about 10 undergraduate students through this program.

**Dr. Harsha Madhyastha** was hired in 2010 to augment our existing strength in networking and to add a more systems-building approach to research and teaching at UCR. Dr. Madhyastha has significantly revised the material for CS 153 to make the material more up-to-date. The class now covers all issues underlying operating systems, with a discussion on how various operating systems components (such as scheduling and virtual memory management) are implemented in modern operating systems such as Windows, Linux, and Mac OS. Dr. Madhyastha has also ensured now that the students in CS 153 get their "hands dirty" implementing the basics of operating systems, rather than simply learning the theory underlying them in class. For this, the class now includes projects based on the popular Nachos framework, which is also used in undergraduate operating systems classes at UC Berkeley and UC San Diego. Moreover, these projects are done in teams and requires students to build on top of a basic Nachos code base. This gives the students experience with working in groups and with adding to legacy code, issues that they will have to deal with when they graduate and take up positions in the industry. Since the Nachos framework requires students to write code in Java, an added benefit of the revised CS 153 is that UCR CSE students gain expertise in a new programming language.

**Dr. Tamar Shinar** was hired in 2010 because we only had one professor (Dr Zordan) who could teach our very popular classes in computer graphics, and in order to strengthen the department's presence in computer graphics and scientific computing. She has given informational lectures about her research in the undergraduate Computer Science seminar, advised undergrads with an interest in pursuing a career in graphics, and will be teaching undergraduate level computer graphics this academic year.

Finally, at the time of writing this document we have permission to hire **two new faculty**. We have completed our interview cycle and we have offers out. We are seeking candidates in Operating/Distributed Systems, Cyber-security and Cyber-physical systems.

1. **Faculty Workload**

The Faculty workload is shown in table 6-2 later in this section.

1. **Faculty Size**

The faculty cover all the broad areas of the program as mentioned above. All required courses are covered by the tenured/tenure-track faculty and lecturers. Faculty are actively involved with the students (examples of faculty leading undergraduate research are listed below). Each faculty holds weekly office hours for the students in the courses. Moreover, our BCOE has an informal “open door” policy, and students are invited to drop by professor’s offices any time the door is open, which it is at least 30 hours for the majority of faculty.

All faculty serve on various committees that look into student evaluation and curriculum. University service consists primarily of involvement in department and college committees, as well as Academic Senate committees (as per the shared governance structure of the University of California). All faculty in the CS Program actively participate in research activities and most of them have large research groups funded by extramural grants and contracts. This requires them to be up-to-date on the latest developments in the field by reading journals, attending conferences, participating in government panels, and reviewing activities.

For ladder-track faculty, the teaching load is relatively light. For full-time lecturers, the teaching load is three courses per quarter, but these are not three unique courses. Typically, in the UCR system we expect a faculty member to devote 40% of his/her time to teaching, 40% to research, and 20% to service. The research and service components in particular afford opportunities to remain abreast of developments in the professor’s research field and in pedagogy.

Many of the faculty have close associations with industry, and these are exploited to expose the students to industry. For example in ENGR 101 the instructors bring in industrial and professional practitioners to both give formal talks, and participate in informal question and answer sessions. In the most recent offering, guest speakers included, Dr. Casey Czechowski (Teradata), Diego Villasenor (Microsoft), Dr. Lisa McIlwain and Dr. Sivaram Gopalakrishnan (Synopsys), Jose Medina (Dreamworks) and Marc Soriano (RCC Norco and Art Institute Santa Monica).

Faculty are eligible for sabbaticals. Faculty members have resources from initial complements, “*various donors*” funds, and contract and grant awards to travel to meetings and conferences in their disciplinary areas or in engineering education. Some additional funds are available from the College, the campus, and the Faculty Senate. These resources are sufficient to assure that professors are able to maintain currency in their fields. New faculty receive startup packages that allow them to purchase excellent facilities. Every four years, each UCR faculty member gets a $2000 stipend to upgrade office computing facilities.

To address faculty’s currency in pedagogy, the UCR Office of Instructional Development has established a Scholarship of Teaching lecture series for faculty and instructors to enhance the quality of teaching throughout the campus. Presentations highlight

* The effective use of current and emerging instructional methodologies and technologies.
* Strategies for the introduction of active learning, peer to peer learning, and collaborative approaches in teaching.
* Pedagogical approaches to enhance student engagement and optimize student learning outcomes.
* Effective approaches to teaching and learning in and outside of the classroom.
* The engagement of teaching community in the collaborative, scholarly examination of their practice as teachers.
* The development of assessment tools to measure student learning outcomes.
* The development of a campus culture of evidence regarding our academic programs.

Not only do many CS faculty *attend* these lectures, but members of the Computer Science department have *taught* them. For example Dr. Victor Zordan taught a lecture on teaching to Multidisciplinary teams. Most lectures are presented by faculty or administrators from UCR, and some by outside presenters. Many deal with new teaching resources and technologies available for use at UCR.

***6.C.1 An Overview of our Efforts and Achievements in Undergraduate Research***

The department has a long tradition of encouraging undergraduate research. For example, research efforts with undergraduates are highlighted in the department’s cover letter for promotion and tenure. In every case the faculty members labs are large enough to allow undergrads desk space next to graduate students and post-docs, and most of our faculty take advantage of this.

**Some Representative Examples of how our Faculty Engages Undergraduate Research:**

In his three years at UC Riverside, Dr. **Iulian Neamtiu** has established a strong track record of supervising and publishing with undergraduate students. He has supervised one undergraduate student (Michael Lambert) under a Title V/STEM project (undergraduate research for Hispanic and low-income students). He has supervised another undergraduate student, Garrett Wong, for more than one year (2010--2011). Garrett is now an MS student at UC Irvine. In Summer 2011 he has supervised two undergraduate students, (one female, Nada Hashem), and one Hispanic (Lorenzo Gomez) on research projects. Nada Hashem is currently applying to MS programs in bioinformatics. Lorenzo Gomez will pursue a PhD in Computer Science; he has been offered admission into the PhD programs at USC, UCI and UCLA.

Dr. **Philip Brisk** is a strong advocate of undergraduate research at UCR. He presently supervises approximately 10 undergraduate student researchers, the most advanced of which (approximately 5) are funded part-time through his NSF grant. The majority of these undergraduate student researchers are working on topics relating to programmable microfluidics, and one is using the Stellarton development board to accelerate image processing algorithms using the FPGA. Several of these students have co-authored papers submitted for publication to top-tier conferences in embedded systems, and several other papers co-authored by undergraduates are in preparation at the time of writing.

Dr. **Eamonn Keogh** demonstrates the decade-long culture of the CS department in encouraging and valuing undergraduate research. In his first year with the department he wrote two papers with an undergrad, Ms. Shruti Kasetty. One of those papers, now has 539 citations (Google Scholar) and is widely regarded as a classic work in time series data mining. Dr Keogh has also published at least one top-tier paper with the following UCR undergrads Shashwati Kasetty, Scott Sirowy, Sam Meshkin, Jin Shieh and Isaac Espinoza. Dr Keogh is currently supervising two undergraduate students, Vinci Sevilla and Dante Jamal O'Hara who are both working on an insect classification project funded by the Bill and Melinda Gates foundation and a gift from IBM. With Dr. Keogh’s help and advice, both students are using this research experience as the cornerstone of their applications to graduate schools, and to apply for funding to cover their last year as undergrads. For example, Vinci has just applied for a UCR Chancellor's Research Fellowship, and Dante has *won* a HSI (Hispanic Serving Institution) Undergraduate Research Award based on the insect classification device he built in Keoghs Lab. Dr. Keogh has won more than $17,000 of grants just for undergrad research.

Dr. **Christian Shelton** has mentored four undergraduate students in his lab over the past five years. One is a current UCR undergraduate who has applied to PhD programs. The other three graduated in 2010. Two are current PhD students (at UCSD and UCI) in computer science, one just finished his MS degree (from UCSD). They are authors on two papers from Dr. Shelton's group, including a collaboration after leaving UCR. Dr. Shelton makes a point to include all undergraduates in group research meetings, to give them equal space in his research lab, and to encourage them to fully participate in the research activities so that they can better understand the research process and whether graduate work is of interest to them. He meets with undergraduate students in his group weekly (just as with graduate students) to discuss progress and possible solutions.

Dr. **Vagelis Hristidis** has worked with several undergraduate students on research projects in the last five years. These students were mostly supported by NSF Research Experiences for Undergraduates supplements. Dr. Hristidis has rigorously followed the process of regular research meetings, where students are exposed to the research process, as well as concrete research/implementation assignments. He has worked with students Alejandro Hernandez and Dionny Santiago, currently working for Ultimate Software in Miami, Michael Tracey, currently at Lockheed Martin, Salma Rodriguez, currently a senior student, and others.

Drs. **Marek Chrobak** and **Neal Young** have been working with undergraduate students on two research projects related to the design of algorithms and data structures. One project was focused on algorithms for drawing diagrams. Of the three students involved in this project, two (M. Yang and M. Ngan), are now in graduate programs, one at UCR and one at USC. The other research projects involved four students: S.Denny, F.Thomas, J.Fang and C.Manghane, who are working on designing efficient algorithms for computing dominators in graphs, a problem that has a variety of applications ranging from compiler design to the design of multi-plot games. S.Denny has graduated last year and entered our graduate program, while the other three students continue working on this project. All of these students were supported by an REU grant from NSF. One of the students (M. Ngan) is female and one (C. Manghane) is African-American.

Dr. **Walid** **Najjar** has always been a very strong advocate of extensive faculty support for undergraduate research, having been himself the beneficiary of such efforts. Over the past four years Dr. Najjar has supervised the research experience of six undergraduate students. All of these are either already enrolled in a graduate program in CSE or have applied to one. These include Adrian Park (co-author on two research papers), Robert Halstead (co-author on two research papers), Joseph Tarango, Xi Luo, Scott Denny and Skyler Windh.

Dr. **Stefano Lonardi** has advised several undergraduate students either in a quarter of “directed studies” and/or by involving them directly in the research of his lab. He has advised eight undergraduate students since 2002. At least four of these students were admitted to graduate school. The last undergraduate student under his supervision, Matt Alpert, worked in his lab for the last two years. Dr. Lonardi was able to obtain NSF REU for him twice during the summer. His contributions to Lonardi’s project on the genome sequencing of barley have been very significant. He is a co-author of a manuscript currently submitted a high impact factor journal and as a result of his work, Matt was awarded the 2011 CEPCEB Undergraduate Research Award, which is a campus-wide recognition at UC Riverside.

Prof. **Frank Vahid** actively seeks undergraduate researchers to work in his embedded systems research lab, to give them experiences that may improve their chances of completing their degree, to better understand and master their course material, and to encourage top students to consider graduate study. Prof. Vahid has consistently employed between 3-6 undergraduates at any given time during the past 15 years. Prof. Vahid presently employs six undergraduate students, with the two seniors planning on graduate studies with Vahid at UCR next year. Last year's two seniors went on to graduate studies in top research groups at UCLA and at EPFL (Switzerland). Eight of his past undergrad researchers obtained PhDs with Vahid, five of those became professors (two female), three of those thus far have received NSF CAREER awards, and one other recently won the prestigious international Terman award for young professors. Of the 20 or so undergraduates employed by Vahid in the past 10 years (about half of them female), nearly all have gone on to graduate studies, with most of those indicating they'd originally had no intention of considering grad school, and with the other students obtaining excellent jobs in part due to their unique lab experiences. In the past 10 years, Prof. Vahid has obtained more than 5 grants specifically to support undergrad research, totaling $150,000.

Dr. **Gianfranco Ciardo** has worked with several undergraduate students at UCR in the past years. Jevons Chen and Miguel Rodriguez spent the 2000 Summer quarter in an internship in Dr. Ciardo's lab, working in conjunction with graduate students on implementation of a software verification tool. Diego Villasenor was affiliated for two years with Dr. Ciardo's lab, supported by a UC LEADS fellowship; he implemented a run-time dynamic visualization tool in java to display large decision diagrams as they are being built. He is now a PhD student at UCLA. Currently, Dr. Ciardo is supervising Mantej Singh Rajpal on a research, and eventually implementation, project aimed at storing very large sets of arbitrary (but finite) length strings, and efficiently manipulating them symbolically (instead of one-by-one).

Dr. **Victor Zordan** has a continuing history of working with undergraduates in his research lab. He averages projects with 2-3 undergraduates per year and many of these have gone on to graduate school. One example is David Brown who has seen good success thus far. As an undergraduate David began volunteering in Dr. Zordan's lab and published a paper as a co-author in a top conference in 2010. He also was admitted into our first cohort for the 5-year BS/MS program. Beyond being a top student academically, he has recently submitted a strong submission to Siggraph and has been admitted to several good schools to pursue his Ph.D. He plans to attend University of British Columbia in the fall. A second example, Raul Arista, (an ME undergraduate) has been working with Dr. Zordan for the past year, he will also be a co-author on a solid submission (in preparation) and is likely to attend graduate school. Other students (CS/CE) have worked with Dr. Zordan and most choose to attend graduate school after the experience.

1. **Professional Development**

All faculty members are expected to be active in research and professional activity throughout their careers. It is common for new faculty hires to have money in their initial complements for travel to meetings of professional societies or other scholarly events. Later in their careers, grant funding typically supports the cost of travel to meetings and conferences, where they share research results.

The departments and degree programs cooperate to present lecture series every academic year. These series bring faculty candidates and distinguished guests from academia or industry to campus. Faculty and students attend these sessions.

Intramurally, professional development opportunities include workshops on teaching skills, interpersonal skills, and other matters. State law and University policy also require training in sexual harassment prevention, laboratory safety, and other matters.

For the past few years, the National Science Foundation has required grantees to provide training in responsible conduct of research (RCR) to all trainees who are paid on NSF grants. In response, UCR and the College have established training resources including an on-line tutorial, and departments are encouraged to include topics in research ethics and engineering ethics in their lecture series and courses. By being required to train their students in RCR, faculty members continually refresh themselves in this subject area. Similarly, NSF requires postdoctoral trainees who are supported by its grants to be mentored by their faculty advisors so they can become independent investigators. This mentoring takes many forms but requires faculty members to maintain their skills as mentors.

Faculty members have significant resources from initial complements, “various donors” funds, and contract and grant awards to travel to meetings and conferences in their disciplinary areas or in engineering education. Many of the faculty aggressively seek, and have been successful in obtaining REU supplements to NSF awards. Additional funds are available from the College, the campus, and the Faculty Senate. One concrete example includes the yearly Academic Senate Award, which faculty may applied for to allow funding for conference travel, undergrad research etc. These awards range in value, but are typically $1,500. While these awards are not guaranteed, the acceptance rate is over 80%. These resources are more than sufficient to assure that professors are able to maintain currency in their fields.

Regular sabbatical leave provides salary at varying percentages of regular salary, depending on the amount of accrued sabbatical leave credit and the option elected by the eligible appointee. However, as a concrete example, a faculty member that has worked 18 quarters is entitled to two quarters sabbatical at full salary. An individual on such regular sabbatical leave is excused from all regular duties to enable full-time effort to research and/or study. The Leaves of Absence/Sabbatical Leave policies are detailed in document APM-740, which is publicly available at this URL:

*www.ucop.edu/acadpersonnel/apm/apm-740.pdf*

The faculty frequently attend and present at conferences and workshops worldwide.

1. **Authority and Responsibility of Faculty**

All program issues are initiated by the Departmental faculty. Many of the faculties devise new courses as technical electives based on their research expertise. Changes in program must be approved by a vote of the Departmental faculty and also the Executive Committee of the College of Engineering as depicted in Figure 21.

|  |
| --- |
|  |

Figure 21: Process for Obtaining Academic Approval for Program/Course Changes

**Note**: Below we describe the role played by the faculty in *course revision*. The role they play *in the development and implementation of the processes for the evaluation, assessment, and continuing improvement of the program, including its program educational objectives and student outcomes* is subsumed by Section 4.

Course creation, modification and evaluation are entirely the responsibility of CS faculty. The Dean of the College (or anyone else) may make suggestions about curricular matters, but it is the faculty’s responsibility to take action.

Courses may be created or modified through an established process through the faculty governance system. Ultimately, the Committee on Courses (a campus wide-committee), has authority for final approval of all courses of the Riverside Division. This committee consists of a minimum of eight members, normally with at least one member representing each of the areas: humanities, social sciences, biological sciences, and physical sciences, and each of the colleges/schools. One member of the Committee on Courses is also a member of the Committee on Educational Policy. In the recent past, Dr. Neal E. Young, Computer Science & Engineering dept has been a committee member, and in the next Academic year, Dr. Philip Brisk, Computer Science & Engineering will serve.

The procedure for course creation or modification is as follows:

1. A department faculty member identifies a need for course creation or modification. If the faculty member is not familiar with the process, he/she will read the following documents, which are maintained at *http://senate.ucr.edu/committee/?do=info&id=8* :

* Preparer's Course Request Checklist and Quick Tips
* Faculty Checklist
* General Rules and Policies Governing Courses of Instruction (Course Guidelines)
* Submission Deadlines for Course Proposals for 2012-13

These documents are *very* detailed and complete, and perhaps a little intimating to a first time proposer. However, Academic Senate support staff member, Ms. Marla Jo Booth is very responsive to requests for help.

1. After discussion with other department faculty, a “*Request to Approve a New Course or Revise a Course*” form is completed.
2. Department faculty members, at every rank, vote on the proposal. At this stage the ABET Binders (c.f. Section 4.B) are consulted.
3. The request is sent to the Committee on Courses, which presents the request at a meeting of the Committee for a vote.

Responsibility for consistency and quality of courses resides within the department.

Table 6-1. Faculty Qualifications

Computer Science

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Faculty Name | Highest Degree Earned- Field and Year | Rank 1 | Type of Academic Appointment2  T, TT, NTT | FT or PT3 | Years of Experience | | | Professional Registration/ Certification | Level of Activity4  H, M, or L | | |
| Govt./Ind. Practice | Teaching | This Institution | Professional Organizations | Professional Development | Consulting/summer work in industry |
| Laxmi N. Bhuyan | Phd: 1981 | P | T | FT | 3 | 23 | 12 |  | M | M | M |
| Philip Brisk | Phd: 2006 | AST | TT | FT | 6 | 3 | 3 |  | M | M | L |
| Marek Chrobak | Phd: 1985 | P | T | FT | 0 | 17 | 15 |  | M | M | M |
| Gianfranco Ciardo | Phd: 1989 | P | T | FT | 6 | 20 | 9 |  | M | M | L |
| Michalis Faloutsos | Phd: 1999 | P | T | FT | 0 | 13 | 13 | ACM/IEEE | L | M | M |
| Rajiv Gupta | Phd: 1987 | P | T | FT | 3 | 22 | 5 | ACM/IEEE | H | M | L |
| Vagelis Hristidis | Phd: 2004 | ASC | T | FT | 0 | 8 | 1 |  | M | M | L |
| Tao Jiang | Phd: 1988 | P | T | FT | 0 | 23 | 13 | ACM | H | M | L |
| Eamonn Keogh | Phd: 2001 | P | T | FT | 0 | 11 | 11 | ACM | M | H | L |
| Srikanth Krishnamurthy | Phd: 1997 | P | T | FT | 3 | 11 | 11 | ACM/IEEE | H | M | L |
| Stefano Lonardi | Phd: 2001 | P | T | FT | 0 | 11 | 11 | ACM/IEEE | M | M | M |
| Harsha Madhyastha | Phd: 2008 | AST | TT | FT | 0 | 2 | 2 | ACM/IEEE/USENIX | M | M | L |
| Mart Molle | Phd: 1980 | P | T | FT | 0 | 21 | 14 |  | M | M | M |
| Walid Najjar | Phd: 1988 | P | T | FT | 0 | 24 | 12 | ACM/IEEE/AAAS | H | M | H |
| Iulian Neamtiu | Phd: 2008 | AST | TT | FT | 3 | 3 | 3 | ACM | H | M | M |
| Chinya Ravishankar | Phd: 1993 | P | T | FT | 5 | 24 | 13 | ACM/IEEE | M | L | L |
| Christian Shelton | Phd: 2001 | ASC | T | FT | 1 | 8 | 8 |  | M | M | M |
| Tamar Shinar | Phd: 2008 | AST | TT | FT | 5 | 1 | 1 | ACM | M | M | L |
| Frank Vahid | Phd: 1994 | P | T | FT | 1 | 19 | 18 |  | M | M | M |
| Neal Young | Phd: 1991 | P | T | FT | 4 | 12 | 8 |  | L | M | M |
| Victor Zordan | Phd: 2002 | ASC | T | FT | 1 | 10 | 9 |  | M | M | M |
| Vassilis Tsotras | Phd: 1991 | P | T | FT | 0 | 21 | 15 |  | L | M | L |
| Thomas Payne | Phd: 1967 | ASC | T | FT | 0 | 44 | 44 |  | M | M | M |
|  | | | | | | | | | | | |
| Ray Klefstad | Phd: 1998 |  | NTT | FT |  |  |  |  | L | H | L |
| Brian Linard | Phd: 1998 |  | NTT | FT |  |  |  |  | L | M | L |
| Kris Miller | Msc: |  | NTT | FT |  |  |  |  | L | M | L |
| Ryan Rusich | Phd: 2010 |  | NTT | FT |  |  |  |  | L | M | L |
| Scott Sirowy | Phd: 2010 |  | NTT | FT |  |  |  |  | M | L | L |
| Victor Hill | Msc: |  | NTT | FT |  |  |  |  | H | H | L |

Table 6-2. Faculty Workload Summary

Computer Science Department

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Faculty Member**  **(Name)** | **FT or**  **PT** | **Classes Taught (Course No. /Credit Hrs.)** | **Total Activity Distribution** | | | | | | **Other service (if applicable)** |
| **Teaching** | | **Research** | | **Other** | |
|  | **Year** |  | **Year** |  | **Year** |
| Laxmi N. Bhuyan | FT |  |  | 20% |  | 50% |  | 30% | CS Chair |
| Philip Brisk | FT | (W11) CS120B, (W12) CS120B, (F11) CS161, CS161L, (W12)179J, (S12) CS120B |  | 20% |  | 50% |  | 30% |  |
| Marek Chrobak | FT | W(11) CS111, (S11) CS111, (F11) CS111, (W12) CS111, (S12) CS111 |  | 20% |  | 50% |  | 30% |  |
| Gianfranco Ciardo | FT | (S11) CS177, (F11) CS179K, (W12) CS150 |  | 20% |  | 50% |  | 30% | CS Grad Advisor |
| Michalis Faloutsos | FT | (F10) ENGR001G, (F11) ENGR 001G,I,M |  | 20% |  | 50% |  | 30% | Publicity chair |
| Rajiv Gupta | FT | (W11) CS152, (W12) CS152 |  | 20% |  | 50% |  | 30% |  |
| Vagelis Hristidis | FT | (S12) CS172 |  | 20% |  | 50% |  | 30% |  |
| Tao Jiang | FT | (W11) CS150, (S11) CS150, (S12) CS150 |  | 20% |  | 50% |  | 30% | Graduate Admissions |
| Eamonn Keogh | FT | (F10) CS179M, (W11) CS005 |  | 20% |  | 50% |  | 30% | ABET Chair |
| Srikanth Krishnamurthy | FT | (S11) CS169, (W12) CS164 |  | 20% |  | 50% |  | 30% | Search Chair |
| Stefano Lonardi | FT | (F10) CS141, (F11) CS141 |  | 20% |  | 50% |  | 30% | Associate Chair |
| Harsha Madhyastha | FT | (S11) CS153, (W12) CS153 |  | 20% |  | 50% |  | 30% | Search Committee |
| Mart Molle | FT | (W11) CS164, (S11) CS30, CS179I,  (W12) CS177, (S12) CS30, CS179I |  | 20% |  | 50% |  | 30% |  |
| Walid Najjar | FT | (F10) CS161, CS161L, (S11) CS161, CS161L, (S12) CS161, CS161L |  | 20% |  | 50% |  | 30% | CEN Program Director |
| Iulian Neamtiu | FT | (W11) CS180, (W12) CS180 |  | 20% |  | 50% |  | 30% | Search Committee |
| Chinya Ravishankar | FT | (F10) CS165, (F11) CS165 |  | 20% |  | 30% |  | 50% | Associate Dean |
| Christian Shelton | FT | (W11) CS170, (W12) CS170, (S12) CS181 |  | 20% |  | 50% |  | 30% | Grad Admissions |
| Tamar Shinar | FT | (S12) CS130 |  | 20% |  | 50% |  | 30% | Grad Admissions |
| Frank Vahid | FT | (F10) CS122A, (W11) CS61, CS179J, (W12) CS120B |  | 30% |  | 50% |  | 20% | Undergrad Committee |
| Neal Young | FT | (F10) ENGR101, (S11)CS141, (F11) ENGR101, (S12) CS141 |  | 20% |  | 50% |  | 30% | Undergrad Adviser |
| Victor Zordan | FT | (W11) CS130, (S11) CS134 |  | 0% |  | 0% |  | 0% | Sabbatical |
| Vassilis Tsotras | FT | (F10) CS166, (S11) CS166, (F11) CS166, (S12) CS166 |  | 20% |  | 50% |  | 30% | Search Committee |
| Thomas Payne | FT | (F10) CS100, (W11) CS153, (S11) CS100, (F11) CS100, (W12) CS100, (S12) CS153 |  | 60% |  | 10% |  | 30% | Emeritus |
|  | | | | | | | | | |
| Sharon Burton | PT | (F10,F11,S11,S12) ENGR 180 |  | 100% |  | 0% |  | 0% | Burton & Graham always *co-teach* ENGR 180 |
| Bonni Graham | PT |  | 100% |  | 0% |  | 0% |
| Toby Gustafson | FT | Does not teach major classes |  |  |  | 0% |  | 0% |  |
| Ray Klefstad | FT | (F10) CS5, CS14, (W11) CS6, CS14, (S11) CS6, CS14, (F11) CS5, (W12) CS6, CS14, (S12) CS6, CS100 |  | 100% |  | 0% |  | 0% |  |
| Brian Linard | FT | (F10) CS10, CS61, (W11) CS5, CS12, (S11) CS12, CS61, (F11) CS6, CS61, (W12) CS12, CS61, (S12) CS12, CS61 |  | 100% |  | 0% |  | 0% |  |
| Richard McHard | PT | (F10) CS6, CS111, (F11) CS12, CS14 |  | 100% |  | 0% |  | 0% |  |
| Kris Miller | FT | (F10) CS10, (W11) CS10, (S11) CS10, CS13, (F11) CS10, (W12) CS10, (S12) CS10, CS13 |  | 100% |  | 0% |  | 0% |  |
| Ryan Rusich | PT | (summer2010) CS10 |  | 100% |  | 0% |  | 0% |  |
| Scott Sirowy | PT | (F11) CS122A |  | 100% |  | 0% |  | 0% |  |
| Victor Hill | PT | (F10) CS183, (F11) CS183 |  | 100% |  | 0% |  | 0% |  |

# CRITERION 7. FACILITIES

**A.** **Offices, Classrooms and Laboratories**

The Bourns College of Engineering occupies Bourns Hall (approximately 105,000 assignable square feet with wet labs, classrooms, and offices), Winston Chung Hall (approximately 104,000 square feet with dry labs, classrooms, and offices), and part of the Materials Science and Engineering Building (approximately 77,000 assignable square feet with wet labs, classrooms, and offices). Bourns Hall opened in 1995. Winston Chung Hall opened in 2001, and the MSE Building opened in 2011.

All departments share classrooms and conference rooms. The assignment of classrooms for each course is made by a joint effort between the Student Affair Office of the Bourns College of Engineering and the Scheduling Office of the Registrar Office. The Student Affair Office requests a room from the Scheduling Office providing the enrollment for the individual class. The centralized Scheduling Office then assigns a room in different buildings on campus with the best availability to accommodate the size of the class. Special request for additional lectures, tutorials, discussions, and examinations can be made by the instructor directly within the College of Engineering. The TA office in Winston Chung Hall sometimes can be used to hold additional tutorials with prior acknowledgment.

The Campus has 60 general-assignment classrooms of varying sizes, each of which is equipped with wireless Internet access, a 3000-lumen video projector connected to a networked PC, and the targets/receivers for wireless audience-response clickers.

**Instructional Laboratories**

The Computer Science program is designed to provide students with extensive experience beginning in their first year of classes. Nearly all courses have an associated mandatory lab component.

There are five general purpose instructional labs located on the first floor of Winston Chung Hall in rooms 127, 129, 132, 133, and 136 in which courses are scheduled that support the CE curriculum. These laboratories run CentOS Linux as their base operating system, and provide access to Windows desktop environment and applications via connection over the LAN to a Windows 2008R2 Terminal Server cluster, Each lab is equipped with 32 desktop PCs with a network printer, and laboratory section sizes are typically 30. Lab sections are scheduled in the range from 8 AM to 9 PM in these labs.

There is an additional computer laboratory located in Winston Chung 226 which provided access to the same software as other labs, but is an open lab where students can go to work at any time of the day, even if other laboratories are all scheduled for courses.

There is also a CS laboratory located in Winston Chung 136 that has specialized equipment including Intel IXP 1200 and 2400 network processor cards, and for specific CE courses is supplied with a range of equipment including oscilloscopes, power supplies, function generators, digital multimeters, and FPGAs. The computers in the laboratory run Linux, and so the fraction of embedded systems software that only runs under Windows and additionally requires direct hardware access is run in a virtual machine environment, currently VMWare. Lab sections are scheduled from 8 AM to 11 PM in this lab.

All of these facilities are accessible 24/7 via card access. Additionally, they are accessible on the Internet – Linux via Secure Shell or NX, and Windows via Terminal Services. Computers, printers, and supplies for the laboratories are paid for via a course materials fee that provides approximately $25K per quarter. This fee was established in 2004 and ensures that technology refresh in the laboratories will be sustained for the foreseeable future.

**B. Computing Resources**

Information technology support, services and facilities are available from several sources for use by the programs of The Marlan and Rosemary Bourns College of Engineering and its students, faculty, and staff:

1. Campus-wide support, services, and facilities are provided by Computing and Communications (C&C) and managed by full-time professional staff.

2. The College, through its programs of Chemical/Environmental Engineering, Computer Science and Engineering, Electrical Engineering, and Mechanical Engineering, and its Research units also provide a variety of technical services and support.

Details of these support, services, and facilities are as follows:

**C&C Overview**

**• Support Services**

**• Facilities and Infrastructure**

**• Other Services and Support**

C&C (which includes the Instructional Technology Group, Computing Infrastructure and Security, the Computer Support Group, and Communications) is under the direction of the Associate Vice Chancellor and CIO who reports to the Provost. The Instructional Technology Group, Computer Support Group, and Communications sub-units have primary responsibility for providing network access and general computing services to the UC Riverside campus.

**Support Services**

**• Instructional Technology Support**

C&C’s Instructional Technology Group offers faculty and students technical and pedagogical support that is academic discipline specific. The Instructional Technology Group emphasize a “hands-on” approach to its services including Blackboard (learning management system) training and support and the management and support of campus site-licensed software.

• **Classroom Technology Support**

C&C provides classroom technology support, services, and infrastructure services (e.g. connection to the wireless network, projection systems, etc.). UCR’s best-of-breed technology-enabled classrooms include the following:

* The capability to present materials from a wide variety of sources, including (at a minimum) DVD, document camera, a personal computer, laptop computer, and Internet.
* Chalkboard or whiteboard that is available and viewable at the same time digital or analog presentations are underway.
* Combination of high-powered data projectors and/or lighting zone controls that allow students to take notes and view presentation material at the same time.
* “Self-service” design which allows instruction to occur without the aid of technical operators and without the delivery of equipment.
* Based on the academic discipline, sound systems and data projection resolution requirements may drive certain classroom minimum standards.

UCR has implemented “clicker” technology in all its classrooms. In actual use on this campus clicker technology has been shown to:

* Increase attendance (sometimes dramatically)
* Coax participation from normally non-participative students
* Create a more engaging lecture environment

Additionally, all UCR classrooms are equipped with podcasting capabilities. This can be in the form of audio podcasting or lecture capture as supported by Echo360 course capture technology. Students in these classrooms will have on-demand access to archived educational content as presented during lecture, including a video camera feed and classroom audio.

**• General Technology Support**

C&C provides UCR faculty and students with technology to assist them in their instructional and academic pursuits. Services like e-mail, iLearn (Blackboard Learning Management System) and the wireless network ensure that all of UCR faculty and students stay connected with their colleagues, peers and the rest of the world. The Computer Support Group provides desktop computing support for faculty and staff. Services include consulting on hardware, software and networking, plus assistance with acquiring, learning and using stand-alone or networked microcomputers (Windows, Macintosh, Linux, and UNIX platforms). Services offered include telephone support, on-site and carry-in services, on-line remote support, a knowledge base and software downloads. C&C also implemented and spearhead the Microcomputer Support Specialist (MSS) program, which provides decentralized departmental support.

**• Multimedia Development and Research Visualization Support**

This group provides innovative and creative full service web and graphic design for the UCR campus and community. With fully integrated, back-end programming solutions tailored to each client's specific needs, the group supports the university's efforts to secure extramural funds and the campus’ various outreach efforts.

**Facilities and Infrastructure**

**• Computer Labs**

Student Computing Services maintains four public computer labs featuring approximately 149 computers available for academic use by all UCR students, with open hours of approximately 160 hours per week. Faculty instructing a course may reserve the public computing facilities for instructional use or request to have software installed on the machines. Lab assistance and software checkout is available in the labs. C&C provides research software (SAS, SPSS, Mathematica) in most public computer labs.

**• Classrooms and Learning Spaces**

The Multimedia Technologies Group maintains all of UCR's general assignment classrooms that have been equipped with data/video projectors, document cameras, DVD players, PC computer on the network, computer interface for laptop users and network connections. Lecture halls are also equipped with wireless microphones and multiple (two to three) projection systems. Their commitment to instructional technology has led the design and implementation of “smarter” classrooms, such as the Flex Rooms and the Hyperstruction Studio. These rooms feature mobile furniture, whiteboards on every wall, and multiple projection systems.

All general assignment classrooms are equipped with a multimedia controller maintained by C&Cs Multimedia Technologies Group for operation of the various presentation technologies and audio equipment. Internet connectivity is via a robust wired and wireless network. Each controller has a “Help” button for the instructor to alert technicians if there is a problem with the equipment.

A help desk is staffed full time, and at least one field technician is available on campus during instructional hours. Either the help desk (working remotely) or the field technician (in the classroom) can quickly resolve any problem that occurs. In a survey (most recently conducted in 2011), 90% of instructors responded that UCR’s available classroom technology either “Completely” or “Mostly” met their pedagogical needs.

**• Research Technology**

As part of UCR’s Cyberinfrastructure (CI) strategy, C&C supports three computational cluster support models. These include departmentally maintained clusters, dedicated clusters, and a shared collaborative cluster. Three programs are described as follows:

1. A centrally managed, standardized/dedicated cluster of processors, in which researchers pay an annual fee for essentially unlimited use.

2. A collaborative computational cluster, in which each PI can buy a certain amount of hardware, which Computing and Communications will manage. The PI has priority access to the equipment that he or she acquired, plus access to the entire cluster as available. UCR’s collaborative cluster provides a shared system as a computing resource for campus researchers with limited financial resources.

3. Departmentally maintained clusters, centrally managed. This type of cluster is meant for researchers who have computing needs that fall outside of the campus cluster standards. These systems are built to particular PI/lab/center specifications and managed by PI funded staff, but housed within C&C’s data center with C&C staff management /

mentoring / backup provided to the departmental administrator

C&C also provides other research technology support, ranging from network creation / configuration, colocation support, budget preparation / equipment configuration, and cloud services provisioning.

**• Wired and Wireless Networks**

UCR supports 1,200+ wireless access points that provide wireless connectivity to approximately 8,000 concurrent users daily. Additionally, the campus network backbone consists of 10 GB fiber-optic connections, with a minimum of 1 GB capacity to each building on campus. The campus has more than 500,000 feet of air blown fiber conduit, which enables the addition of fiber connectivity essentially “on demand”.

**Other Services and Support**

**• Libraries**

* The UCR Libraries have over 400 public computers among the four campus libraries with selected information resources and software to support and enhance student learning and the research and scholarship activities of the University. Specialized software has been installed on the Learning Commons Computers located in Rivera Library 1st Floor, Rivera Basement, Rivera 2nd Floor, Rivera 3rd Floor, and Science Library 1st Floor. 20 wireless laptops/netbooks are available in Rivera and Science Libraries to faculty, students and staff.

**• CENIC Regional Higher Education Network**

* C&C provides support and maintenance of off-campus network access via connections to the CENIC regional higher education network. All Bourns College of Engineering computing facilities and faculty have high-speed access to CENIC members (e.g. other UC campuses, private research universities in California, the California State University System, etc.) and to Internet2 via C&C support of the CENIC network.

**C. Guidance**

Most computer science labs are offered in labs that have just computer workstations, and thus offer few safely challenges of any kind. For the “hardware” classes, the students receive guidance on basic equipment use with equipment orientation documents and help from the TAs.

D. Maintenance and Upgrading of Facilities

BCOE budgets approximately $300,000/year for instructional equipment acquisition and upgrades. These funds are allocated to BCOE academic programs on an annual request basis. Table 32 shows a listing of the instructional equipment obtained by this process over the last three fiscal years can be found on the following list.

Table 32: A listing of the CS instructional equipment obtained in the last three years.

|  |
| --- |
| Lab Chairs  Lab Tables  Windows Terminal Server license  Remark Office OMR upgrade  Supermicro Server Chasis  AMD Server Processors  Intel SATA drives  Hitachi 3TB hard drives  8 GB RAM  Kodak i1220 scanner  MSDN AA License  File Server support contract renewal  File Server  RBC27 Batteries  Barebones server  Mac Workstations  RAM, harddrives, displays, licenses, etc  Projector mounts  Supplmental NICS |

Upgraded hardware must meet the minimum specifications for the courses/labs taught, software requirements, and hardware interaction. Software upgrades must work with the current labs being taught. Additionally, computers must meet the minimum hardware requirements as required by the software. Both hardware & software upgrades are done in a non-disruptive process. This usually occurs during summer before the beginning of the fall quarter.

E. Library Services

Library collections that support the Bourns College of Engineering are housed in the Orbach Science Library. The Orbach Science Library has a seating capacity of 1,500 including individual carrels, study tables and 25 group study rooms. The library makes available 79 computer workstations for students to use in their research and study, and another 32 computers to support information literacy instruction. The entire UCR library system provides both wired and wireless access to the internet for student laptop use, and laptops are available for check-out at the Circulation Desk.

Normal library hours during the regular school year are as follows:

Monday-Thursday 7:30am – 11pm

Friday 7:30am to 5:00pm

Saturday Noon to 5:00pm and Sunday 1:00pm to11:00pm.

The Orbach Science Library maintains a professional staff of eight librarians, all of whom provide reference and research assistance to engineering students, faculty, and staff. Of these librarians, one is assigned subject responsibility for engineering and is available to assist students, faculty and staff with in depth research questions. The Engineering Librarian and Subject Specialist also offers tutorials and classes on engineering information topics, and maintains Web pages and path-finders to assist engineering students, faculty, and staff in locating the information they need.

The UCR Libraries offers a full range of reference services, including walk-up, telephone, and 24/7 e-mail reference services (*Ask A Librarian*) through a UC-wide and national network as well as reference by appointment. The Orbach Science Library reference desk is staffed 52 hours per week during the academic year (9am-8pm. Monday-Thursday, 9am-5pm on Friday) and 40 hours per week during inter-session periods. In addition to these standard services, engineering students can receive additional reference help from other reference librarians who are assigned to the Science Information Services desk. The Engineering Librarian is available for extended consultation on Senior Design or other research projects.

Incoming freshmen typically receive library orientation sessions in their introductory classes. They might also have additional information literacy instruction in classes that require independent research, such as senior design classes. One-on-one or group tutorials are available for any research topic that might be desired and helpful to engineering students.

**Library Collections**

* **Books**

Engineering books are acquired as part of the Orbach Science Library’s purchasing profile, ordered from catalogs or suggested by students, faculty, and staff. Within the past three years, the library has initiated the purchase of engineering e-books and currently supports and maintains a collection of thousands of electronic books in the discipline. The Libraries provides licensed access to all of the current Springer books online, many of the e-books from the CRC EngNetBase, the Knovel Collection, the Wiley Online collection and many more.

Recently, through a special competitive initiative, the UCR Libraries has brought to our campus, from its former Berkeley location, the extensive and world class Water Resources and Archives Collection (WRCA) containing many materials relevant to dam and bridge construction which is also available to engineering students and researchers from across UCR and the UC system.

* **Journals**

The Libraries currently subscribe to 121 engineering print journals, and Engineering students have access to a vast collection of online journals (94,770 unique titles). UCR maintains access, for example, to all of the journals and proceedings of IEEE, OSA, MRS, and ACM, as well as either proceedings or journals from many other societies. Faculty, staff, and students may suggest new books, journals or other media to be purchased by the library. Library users may request materials that are not available on campus through Interlibrary Loans, and the materials will be made available to them at no cost in a very reasonable amount of time.

* **Research (Journal Article) Databases**

UC Riverside engineering students have access to a number of journal databases to assist them in their research in engineering and in other areas of study. Through co-investments with the other eight UC campuses and the California Digital Library (CDL) Inspec, Compendex, and the Web of Science as well as SciFinder Scholar for chemistry and chemical engineering and Biosis or MEDLINE for biotechnological literature are all available to engineering faculty and students. UCR also licenses Water Resources Abstracts locally with the arrival on our campus in 2010 of the Water Resources Archives and Collections.

* **LIBRARY COLLECTIONS**

|  |  |  |
| --- | --- | --- |
|  | **Books** | **Periodicals** |
| Entire Institutional Library | 2,810,229: (PrintVols.) 404,191: (e-Books)  Total Vols.: 3,214,420 | 6,329 (Active Local Titles) |
| Engineering and Computer Science | 71,757 Print / 29305 online | 168 print / 3976 online |

**LIBRARY EXPENDITURES (See key below)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **2008-2009** | **2009-2010** | **2010-2011** |
| Expenditures for Engineering (Total) | $75,749 | $75,107 | $45,975 |
| Print Books | $13,264 | $11,824 | $9,629 |
| [[6]](#footnote-6)Local Costs Only for Engineering Periodicals Subscriptions | $47,589 | $47,706 | [[7]](#footnote-7)$21,163 |
| E-Book Packages (EngNetbase, O’Reilly) | $7,043 | $7,332 | $6,483 |
| [[8]](#footnote-8)Research Databases | $15,185 | $14,741 | $15,957 |

1. **Overall Comments on Facilities**

BCOE follows the University of California Policy on Management of Health, Safety and the Environment and partners closely with UCR’s Office of Environmental Health & Safety, UC Police Department, and UCR Office of Risk Management, and system-wide laboratory best practices to ensure student, faculty, and staff safety while also protecting the environment and BCOE resources.

Each BCOE department has assigned a Laboratory Safety Officer (LSO). The LSOs assist with class lab operations and equipment management, with their departments with development and implementation of the department Chemical Hygiene Plan, and perform periodic laboratory safety audits (at least annually). BCOE LSOs meet monthly to discuss strategy, share lessons learned, and ensure safety in learning and research.

# CRITERION 8. INSTITUTIONAL SUPPORT

1. **Leadership**

The Computer Science Program is led by a Chair (Prof. Laxmi Bhuyan) and an Associate Chair (Prof. Stefano Lonardi).

### B. Program Budget and Financial Support

**B.1 Resources Provided to the Program**

The program is supported by staff, part-time student assistants, teaching assistants, readers, and graders as needed to support individual courses and program administration. The College provides Student Advisors who interact with program students, monitor academic progress, enable registration, and direct them to appropriate services on campus for tutoring, career counseling, etc. Tutoring service is provided at the Learning Center and in the student dormitories (free for students living on campus). The College has developed a Professional Milestones Program to enable each program student to prepare for internships, job interviews, and research opportunities.

The College provides funds to support teaching assistants, graders, and readers, assigned based on course enrollment and need for laboratory supervision. Teaching Assistants conduct discussion sessions in which students are exposed to additional problems and concepts to reinforce material covered in lectures, and to enable students to complete course assignments. All instructors and teaching assistants maintain posted office hours for assisting students outside scheduled classes. The program has a designated Undergraduate Advisor (currently Dr. Neal Young) to oversee curricular matters and to offer advice on curricular issues.

**B.2 Budgeting**

The University of California, Riverside has a multi-step budget development process. The major steps in the annual process are:

February: Campus Budget Call Letter is distributed and meetings held with academic units to discuss faculty renewal models

March: Comprehensive Planning Documents are submitted to the Executive Vice Chancellor

April: Individual unit hearings with senior UCR management

May: Input and feedback from Faculty Senate Committee on Planning and Budget to EVC

June: Final unit budgets announced

All BCOE academic programs receive Permanent University funding for tenure track faculty, program staff, materials and supplies and travel. Table 33 summarizes Permanent University funding allocations to BCOE departments over the last five fiscal years.

**Table 33: College of Engineering 5-year PERM Budget History**

|  |
| --- |
|  |

In addition, BCOE academic departments receive Temporary University funding each fiscal year for lecturers, teaching assistants, instructional equipment, etc. The amounts of these annual allocations over the last five fiscal years can be found in Table 8-2. (Note: FY 11/12 allocations for Instructional Equipment will be made at the end of the fiscal year).

### C. Staffing

The total headcount of administrative, instructional and technical staff in BCOE for FY 11/12 can be found in Appendix D2.

Several years ago, BCOE centralized the following functions in the Dean’s Office: undergraduate student affairs and advising; contract/grant pre-award processing and academic personnel. All other administrative functions (purchasing, payroll, grad student support, etc.) are provided at the departmental level. Over the past five fiscal years, the number of BCOE administrative and technical staff has decreased by 8.75 FTE due to UCR budget reductions. However, all but 0.25 FTE of these positions have occurred in central Dean’s Office operations and were accomplished with little direct impact on BCOE’s academic programs.

During each fiscal year, BCOE administrative and technical staff salaries are compared with salaries of similar positions within BCOE and within other UCR academic and administrative units. Any significant salary lags are addressed through UCR’s staff equity and reclassification process. During the past two fiscal years, 10-11 staff reclass/equities were processed per year. This process has helped to reward and retain experienced BCOE staff.

In addition to offering on-line and in-class training required to perform a staff position’s basic responsibilities (i.e., payroll, purchasing, etc.), UCR offers extensive career development training programs including:

* Certificate programs in Building Core Competencies, Diversity Training, Performance Management, Professional Academic Advising, Professional Graduate Student Advising and Work Leadership
* Emerging Leader (mentorship) Program
* Management Skills Assessment Program

Most of the above training is at no cost to the employee. All required and optional training is offered through UCR’s Human Resource’s Learning Center. The completion of employee’s required and optional training is recorded in UCR’s automated Learning Management System (LMS).

**D. Faculty Hiring and Retention**

BCOE is still growing toward its target size of approximately 120 faculty members, so, even despite budget pressures; faculty recruitment is an annual event. The basic faculty hiring process is as follows:

1. Each year, departments are asked to submit a faculty recruitment plan that is consistent with their strategic plan.
2. The recruitment plan is sent to the Dean for his review.
3. The Dean then outlines a collective recruitment plan for the College and requests ladder-rank faculty lines from the Provost.
4. The Provost makes an allocation of ladder-rank faculty lines to the College and the Dean determines the overall priorities for the College.
5. The Dean lets the departments know if they can begin a search for faculty members and, if so, how many.
6. The department then forms a faculty committee to prepare a detailed recruitment plan for the position(s). The detailed recruitment plan includes a listing of the search committee, written ads and where they will be placed, flyers for distribution at professional conferences, letter templates for bulk mailings to other relevant departments, an affirmative action plan, and a deadline for priority recruitment.
7. Those detailed plans are sent to the Dean, Provost, and Affirmative Action offices for approval.
8. Once approved, ads are placed, mailings are sent, and the College on-line recruitment website is opened. All applications are received through the College recruitment website.
9. All applications received by the priority deadline are reviewed by the faculty search committee. The committee assesses how well the applicants meet the goals of the department and their potential as a faculty colleague.
10. An initial short-list is developed, then further refined until a list of interviewees is developed.
11. Once the list of interviewees is developed, the list is shared with the department at large, the Dean, and the Affirmative Action office. The Affirmative Action office requires reasons for why candidates were not considered for further consideration.
12. Once the department, Dean, and Affirmative Action Office approve the list, the candidates are invited to campus for an interview where they give one or two seminars, meet with department and other potentially relevant faculty, and the dean.
13. Following the interviews, the department recommends one or more candidates to the Dean for approval to make an offer of appointment.  Upon his approval, the candidates are informed of the offer.
14. The offer is contingent upon approval through the campus policies (Academic Personnel Manual and the Call) for faculty appointments. Procedures differ depending on level of appointment.
15. Once a formal offer is signed and approved by the Chancellor, the candidate becomes a faculty member in the department.

**D. 2 Faculty** **Retention**

The primary strategy is to maintain an atmosphere conducive to achieving excellence in all that we do. We strive to recognize excellent performance in teaching, research and service. We provide sufficient resources for the faculty to advance their research: initial complement funds, laboratory space, and assigned students. Annual training is provided for improving teaching methods. The faculty is encouraged to take online training on a regular basis in topic areas such as Health and Safety, Information Security, Leadership, Effective Use of Advanced Technology in the Classroom, etc. They are given assignments to college and campus committees to provide service and growth of responsibilities. We work to accelerate promotion opportunities for outstanding performance. Junior faculty are provided with mentoring by senior faculty members and provided opportunities for them to mentor students.

We want our faculty to be of the highest quality and thereby attractive to other engineering schools. If as a result a faculty member receives an offer from another institution we provide matching offers to retain the individual. These strategies and actions are predominately successful.

### E. Support of Faculty Professional Development

Faculty professional development funds are provided to assistant professors as part of their faculty start-up packages. In addition, the Academic Senate provides travel assistance grants, and the campus provides grants to support innovative teaching. Also, funds are available to all faculty from their faculty support accounts, which are funded by a number of activities including a (small) portion of indirect costs generated by grants and contracts.

The University offers leaves of absence with pay to attend professional meetings or other University business in addition to its normal sabbatical leave program in order to maintain faculty currency. The University also offers other types of leave with or without pay that may extend over a longer period of time, for good cause. The University Leave policies are covered in section V. (Benefits and Privileges) of the Academic Personnel Manual (APM) http://www.ucop.edu/acadpersonnel/apm/sec5-pdf.html.

The College provides funds to cover the cost of the faculty member’s replacement while on leave. Faculty are also given latitude to modify class schedules/exams to some extent when necessary to accommodate specific professional development needs that require short or intermittent absences during the academic year. In some cases, other department faculty assist with covering a particular class or exam.

# PROGRAM CRITERIA

The CS Program covers all the requirements of the Program Criteria. This was addressed line-by-line in great detail in Criterion 5 and thus not repeated here.

The only program criteria not covered explicitly elsewhere is “**Faculty:** Some full time faculty members must have a Ph.D. in computer science”. We note that all twenty-two of our tenure-track faculty have Ph.Ds. Four out of six of our lecturers have Ph.Ds

**APPENDICES**

# Appendix A – Course Syllabi

To prevent this file from becoming unwieldy, this appendix is in a separate file entitled *Appendix\_A\_Course\_Syllabi.pdf*

# Appendix B – Faculty Vitae

To prevent this file from becoming unwieldy, this appendix is in a separate file entitled *Appendix\_B\_* *Faculty\_Vitae.pdf*

# Appendix C – Equipment

Major pieces of equipment used by the program in support of instruction.

**Equipment Managed by the Computer Science and Engineering Department**

|  |  |  |  |
| --- | --- | --- | --- |
| **INSTRUMENT/MANUFACTURER** | **LABS** | | **Quantity** |
| AS-2042G-6RF/TRF SUPERMICRO SERVER | BOURNS | B260 | 1 |
| 6026T-NTR+/3RF SUPERMICRO SERVER | BOURNS | B260 | 5 |
| MSDNAA – AVAILABLE TO ALL STUDENTS ENROLLED IN CSE COURSES by MICROSOFT - ALL TITLES EXCEPT OFFICE | CHUNG | 106 | N/A |
| SHI SOFTWARE; MICROSOFT TITLES - MS OFFICE, VISUAL STUDIO | CHUNG | 106 | N/A |
| 6026T-NTR+/3RF SUPERMICRO SERVER | CHUNG | 106 | 2 |
| SuperServer 6013A-T SUPERMICRO SERVER | CHUNG | 106 | 2 |
| AS-2042G-6RF/TRF SUPERMICRO SERVER | CHUNG | 106 | 1 |
| TRANSPORT GX28 B2881 TYAN SERVER | CHUNG | 106 | 6 |
| PowerEdge1750 DELL SERVER | CHUNG | 106 | 1 |
| FAS250 NETWORK APPLIANCES SERVER | CHUNG | 106 | 2 |
| ASUS SERVER | CHUNG | 106 | 3 |
| OPTIPLEX GX620 DELL DESKTOP COMPUTERS | CHUNG | 110 | 12 |
| 4250DTNSL HP PRINTER | CHUNG | 110 | 1 |
| fi-5120C FUJITSU SCANNER | CHUNG | 110 | 1 |
| i320 KODAK SCANNER | CHUNG | 110 | 1 |
| CanoScan LiDE 70 CANON SCANNER | CHUNG | 110 | 1 |
| OPTIPLEX GX620 DELL DESKTOP COMPUTERS | CHUNG | 118 | 2 |
| 2012 AUTODESK ENTERTAINMENT CREATION SUITE (30 seat license) | CHUNG | 127 | N/A |
| PRECISION T3500 DELL DESKTOP COMPUTERS | CHUNG | 127 | 33 |
| 4250DTNSL HP PRINTER | CHUNG | 127 | 1 |
| PRECISION T3500 DELL DESKTOP COMPUTERS | CHUNG | 129 | 33 |
| 4250DTNSL HP PRINTER | CHUNG | 129 | 1 |
| PRECISION T3500 DELL DESKTOP COMPUTERS | CHUNG | 132 | 32 |
| 4250DTNSL HP PRINTER | CHUNG | 132 | 1 |
| PRECISION T3500 DELL DESKTOP COMPUTERS | CHUNG | 133 | 32 |
| 4250DTNSL HP PRINTER | CHUNG | 133 | 1 |
| PRECISION T3400 DELL DESKTOP COMPUTERS | CHUNG | 135 | 33 |
| 4250DTNSL HP PRINTER | CHUNG | 135 | 1 |
| PRECISION T3500 DELL DESKTOP COMPUTERS | CHUNG | 136 | 20 |
| 4250DTNSL HP PRINTER | CHUNG | 136 | 1 |
| PRECISION T3400 DELL DESKTOP COMPUTERS | CHUNG | 226 | 32 |
| 4250DTNSL HP PRINTER | CHUNG | 226 | 1 |
| REMARK OFFICE OMR by GRAVIC SOFTWARE - MODEL 8 | CHUNG | 311 | N/A |
| REMARK OFFICE OMR by GRAVIC SOFTWARE - MODEL 8 | CHUNG | 312 | N/A |
| REMARK OFFICE OMR by GRAVIC SOFTWARE - MODEL 8 | CHUNG | mobile | N/A |

# Appendix D – Institutional Summary

### The Institution

#### University of California, Riverside (Legal name: The Regents of the University of California) 900 University Avenue Riverside, CA 92521

#### Chief executive officer of the institution.

|  |  |
| --- | --- |
|  | Timothy P. White, Chancellor |

1. The individual submitting the self-study report.

|  |  |
| --- | --- |
|  | Reza Abbaschian, Dean, Bourns College of Engineering |

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

The University of California, Riverside, is accredited by the Western Association of Schools and Colleges (WASC). UCR was most recently accredited on March 3, 2010. WASC reaccreditation occurs approximately every 10 years, and UCR’s next proposal for reaccreditation is due to be submitted to WASC in fall 2016.

Other accreditations at UCR include:

* Graduate School of Education, accredited by the California Commission on Teacher Credentialing. Reaccreditation is under way now; a report is due in fall 2012, and the next site visit is expected to be in 2014. Further, the GSOE School Psychology program is being reaccredited in 2012. A site visit was in March 2012, and a decision is due in August 2012.
* The Chemistry Department is reviewed by the American Chemical Society. The Chemistry department provides annual reports and 5-year reports on curriculum and student performance. The most recent 5-year report was in June 2010.
* The School of Business Administration (SoBA) will begin its AACSB Maintenance of Accreditation in 2012-13, with a site visit expected in January 2013.
* The UCR School of Medicine was denied initial accreditation by the Liaison Committee on Medical Education (LCME) in June 2011 because of budget uncertainties. The University expects to reapply this year with a new funding model that is less reliant on state funds.

1. **Type of Control**

The University is a state-controlled institution of higher education and an accredited Hispanic Serving Institution (HSI).

1. **Educational Unit**

The following chart describes the program organizational structure for the Bourns College of Engineering. Each program chair reports to the Dean of the College, who reports to the Vice Chancellor and Provost, who reports to the Chancellor of the UC Riverside Campus. The program chairs shown on the top line of the college section are also Department Chairs. The Computer Engineering Program is supported by faculty from both the Electrical Engineering and Computer Science Programs. The Material Science and Engineering Program includes faculty from the Bioengineering, Mechanical Engineering, Chemical Engineering, Environmental Engineering, Electrical Engineering, and Computer Science Programs.

|  |
| --- |
|  |

1. **Academic Support Units**

Table 34 lists the names and titles of the individuals responsible for each of the units that teach courses required by the program being evaluated.

Table 34: Academic Support Units

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Chemistry**  *eric.chronister@ucr.edu*  951-827-3288 | Eric Chronister | *Chair* |
| L%20Bhuyan%20at%20200 | **Computer Science**  *bhuyan@cs.ucr.edu*  951-827-2244 | Laxmi Bhuyan | *Chair* |
|  | **Electrical Engineering**  *farrell@ee.ucr.edu*  951-827-2159 | Jay Farrell | *Chair* |
| willis | **English**  *deborah.willis@ucr.edu*  951-827-1458 | Deborah Willis | *Chair* |
|  | **Math**  *chari@math.ucr.edu*  951-827 6463 | Vyjayanthi Chari | *Chair* |
|  | **Physics**  *jory.yarmoff@ucr.edu*  951-827-5336 | Jory Yarmoff | *Acting Chair* |
|  | **Statistics**  *daniel.jeske@ucr.edu*  951-827-3014 | Daniel Jeske | *Chair* |

1. **Non-academic Support Units**

|  |  |
| --- | --- |
| http://asp.ucr.edu/SiteCollectionImages/Faculty/Ruth Jackson.jpg | UCR Libraries: Dr. Ruth Jackson, University Librarian  *ruth.jackson@ucr.edu* |
| http://cnc.ucr.edu/images/about_mgmt/Chuck_Rowley_05.jpg | Computing & Communications: Charles J. Rowley, Associate Vice Chancellor & Chief *Information Officer*  *rowley@ucr.edu* |
| http://ue.ucr.edu/ue/img/staff/wong.png | Learning Center: Michael P. Wong, Director  *michaelpaul.wong@ucr.edu* |
|  | Career Center: Randy Williams, Director  *randy.williams@ucr.edu* |

1. **Credit Unit**

One quarter credit represents one class hour or three laboratory hours per week. One academic year normally represents 30 weeks of classes, exclusive of final examinations.

1. **Tables**

See below.

Table D-1. Program Enrollment and Degree Data

**Computer Science & Engineering**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Academic Year | | Enrollment Year | | | | | Total  Undergrad | Total  Grad | Degrees Awarded | | | |
|  | 1st | 2nd | 3rd | 4th | 5th | Associates | Bachelors | Masters | Doctorates |
| Current | 2011-12 | FT | 94 | 102 | 49 | 39 | 26 | 310 | 154 | N/A |  |  |  |
| Year | PT | 2 | 1 | 0 | 0 | 7 | 10 | 2 |  |  |  |
| 2010-11 |  | FT | 129 | 62 | 42 | 41 | 19 | 293 | 141 | N/A | 37 | 18 | 8 |
|  | PT | 2 | 0 | 3 | 0 | 6 | 11 | 3 |  |  |  |
| 2009-10 |  | FT | 85 | 58 | 40 | 32 | 22 | 237 | 145 | N/A | 36 | 18 | 17 |
|  | PT | 1 | 2 | 3 | 0 | 9 | 15 | 7 |  |  |  |
| 2008-09 |  | FT | 100 | 57 | 41 | 26 | 33 | 257 | 132 | N/A | 42 | 14 | 15 |
|  | PT | 1 | 1 | 2 | 2 | 10 | 16 | 5 |  |  |  |
| 2007-08 |  | FT | 85 | 53 | 34 | 38 | 34 | 244 | 113 | N/A | 57 | 17 | 21 |
|  | PT | 0 | 1 | 1 | 4 | 7 | 13 | 4 |  |  |  |

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

FT--full time

PT--part time

**Table D-2. Personnel**

Year: Fall 2011: Computer Science and Engineering Department

|  |  |  |  |
| --- | --- | --- | --- |
|  | HEAD COUNT | | FTE |
|  | FT | PT |
| Administrative | 0 | 0 | - |
| Faculty (tenure-track) | 21 | 2 | 22.33 |
| Other Faculty (excluding student Assistants) | 9 | 4 | 10.46 |
| Student Teaching Assistants | 15 | 26 | 28.00 |
| Student Research Assistants | 40 | 26 | 53.00 |
| Technicians/Specialists | 2 | 1 | 2.02 |
| Office/Clerical Employees | 3 | 34 | 10.48 |
| Others | 2 | 0 | 2.00 |

# Signature Attesting to Compliance

By signing below, I attest to the following:

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

**Reza Abbaschian**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature Date**

1. All GPAs discussed in this document are out of 4.0 [↑](#footnote-ref-1)
2. GROWL is the secure student portal used to complete the majority of administrative transactions needed during a student’s academic career.  This includes submission of a student’s Statement of Intent to Register, control of all privacy through FERPA based controls, access to their bill, submission of payment, term registration, review of administrative or advising holds, grades, transcript requests and review and acceptance of their financial aid to name a few.  [↑](#footnote-ref-2)
3. The term “*Inland Empire*” is most commonly used in reference to the U.S. Census Bureau's federally defined Riverside-San Bernardino-Ontario metropolitan area, which covers more than 27,000 square miles (70,000 km2). Informally it includes Palm Springs and Palm Desert. [↑](#footnote-ref-3)
4. We note that the assessment and improvement processes are very similar for both the *PEOs* and for program learning *outcomes*, and these two processes run in parallel. This is a deliberate decision to simplify the process and ensure compliance from all parties. The *assessment* part of this process is described in more detail in Criterion 4. [↑](#footnote-ref-4)
5. *www.aacu.org/leap/documents/2009\_EmployerSurvey.pdf*  [↑](#footnote-ref-5)
6. This figure does not include the total amount ($2.4 million ) expended annually by the UCR Libraries as co-investments with other UC campuses and the California Digital Library (CDL) to support access to e-journals, e-books, and electronic databases. The value of the e-journals for supporting engineering alone is over a million dollars annually [↑](#footnote-ref-6)
7. This figure reflects a major journal cancellation which included duplicate and low use titles especially targeting print titles that duplicated e-journal titles. This was a UCR project in response to budget reductions [↑](#footnote-ref-7)
8. Cost for Compendex and Inspec databases. Other databases such as SciFinder, Water Resources Abstracts, Web of Science support multiple disciplines, in addition to Engineering. [↑](#footnote-ref-8)