ABET
Self-Study Report

for the

Computer Science and Engineering Program

at

The Ohio State University

Columbus, Ohio

June 2017

Program Self-Study Report for
Computer Science and Engineering

Contents

Program Self-Study Report
for
CAC/EAC of ABET
Reaccreditation

Background Information

A. Contact Information

Primary contact:	Dr. Neelam Soundarajan Chair, Undergraduate Studies Committee Computer Science and Engineering Dept. Tel: (614) 292 1444 FAX: (614) 292 2911 E-mail: soundarajan.1@osu.edu
Additional contact:	Prof. Xiaodong Zhang Chair Computer Science and Engineering Dept. Tel: (614) 688 3029 FAX: (614) 292 2911 E-mail: zhang.574@osu.edu
Additional contact:	Dr. Rephael Wenger Associate Chair Computer Science and Engineering Dept. Tel: (614) 292 6253 FAX: (614) 292 2911 E-mail: wenger.4@osu.edu
Address:	Computer Science and Engineering Dept. The Ohio State University 2015 Neil Avenue Columbus, OH 43210

B. Program History

The *Bachelor of Science in Computer Science and Engineering (BS-CSE)* program has existed in its current form since the mid-nineties and has been accredited by the EAC and the CAC since 1999[1]. The last general review of the BS-CSE program was in 2011.

One major change was in 2012 when Ohio State moved from the *quarter calendar* to *semesters*, effective

[1]The *BS-CSE* program is in the *Computer Science and Engineering Department* in the *College of Engineering*. The department also offers two other programs, a *BS in Computer and Information Science (BS-CIS)* degree and a *BA in Computer and Information Science (BA-CIS)* degree, both in the *College of Arts and Sciences*. Neither of those programs is accredited by ABET nor are they being currently evaluated. Only the *BS-CSE* program is being evaluated.

Summer, 2012. In preparing for the change, our approach was to retain the essential strengths of the then-existing program while introducing carefully considered revisions based on extensive discussion among faculty and results of some of our assessments. As an example of the former, we decided to continue offering a variety of capstone design courses –rather than a single course as some programs do– so that students interested in different areas of computing can fully pursue their interests by engaging in an intense design project in the specific areas. As an example of the latter, results from our *exit survey*, one of our assessment tools, showed that while students were mostly satisfied with the conceptual focus of our introductory *software engineering* sequence, many were unsatisfied with the use of RESOLVE/C++, a discipline of software development in C++ that is related to research conducted here at OSU. Many students expressed a preference for learning the concepts we taught in the sequence with a different programming language as the delivery mechanism; i.e., a language used more commonly in industry positions of the sort they are most likely to secure for internships and after graduation. Thus, in the semester version of the sequence (CSE 2221, 2231), the conceptual focus of the introductory software courses has very much remained, but we switched to using Java as the programming language and focus more attention on current industry best practices. Details of the process used in designing the semester program were described in an appendix to the self-study we submitted in June 2011 (and will be available during the team's site-visit).

One change that we made during the transition to semesters was the elimination of the requirement that each student complete one of several *technical elective options*, ranging from *software systems*, to *information assurance*, *AI*, etc. Each option listed a number of CSE (and, in some cases, a related field) courses relevant to the particular option that the student was required to include among her elective courses for her program. One of the technical elective options was called the *individualized* option. This allowed students, in consultation with their faculty advisors, to tailor unique combinations of elective courses to meet their particular technical interests which may not fit exactly with any of the specified options. In the transition to semesters, we decided to eliminate all of the specified options and allow each student to choose an appropriate set of elective courses based on her particular interests; in other words, effectively, have all the students follow the individualized option. As it turned out, however, while this ability to tailor unique combinations of technical elective courses was appropriate for some students, for many others, the guidance provided by the other options that we had under the quarter system was indeed very useful; it allowed students, fairly early in their program, to plan out their future courses, based on their technical interests. Perhaps equally importantly, an employer interested in hiring an intern or a full-time employee for a specific position and interested in students/graduates with matching interests could identify them simply on the basis of their tech-elective option (assuming it was not "individualized"!), rather than having to carefully go through the students' course records. Hence, we recently reintroduced *specialization options*, similar to but somewhat more flexible than the technical elective options under the quarter system, with each option requiring students to include, in their respective programs, an appropriate set of CSE (and in some cases related field) courses appopriate to their chosen specialization option. More details of the specialization options appear in subsection (C) below with more complete details later in the self-study under *Criterion 5. Curriculum*, page 40.

Another change that may be worth noting has to do with the courses related to analog circuits and digital logic that our students are required to take. Under the quarter system, the curriculum included 13 quarter-credit hours of courses related to these topics. During the transition to semesters, the Electrical and Computer Engineering (ECE) Dept. which teaches these courses reorganized them and part of some other courses (taken by ECE majors but not part of the BS-CSE program) into two 4 (semester) credit hour courses. The resulting sequence, ECE 2000, 2100 dealt with not just analog circuits and digital logic but also signal processing. While the signal processing material was of some interest to a small number of CSE majors, for most it was not directly relevant. We started receiving negative feedback about the sequence from the student representatives on our Undergraduate Studies Committee as well as the broader student population during our Annual Forum; in essence, the feedback was that portions of each of these courses were very inaccessible and irrelevant to many CSE majors. ECE faculty had also, in the meantime, received negative feedback about the sequence from ECE majors (although the details were somewhat different from the feedback from CSE majors). In any case, the ECE faculty went through extensive discussions, including input from the CSE Undergraduate Studies Committee, and came up with a completely revised sequence of courses. In the revised version, CSE majors take two 3-credit hour courses, ECE 2020 and 2060, the first devoted to analog systems/circuits, the second devoted to digital logic; additional courses on the signal processing-related material are available and may be taken as technical electives by interested CSE majors.

As the field of computing continues to evolve, we have developed new courses and revised existing courses to enable our students to get an understanding of important new ideas and techniques. One important example is CSE 5914, one of our capstone design courses. Originally, the main project in this AI-focused course was based on *expert systems*. But, over the last several years, *machine learning*-based approaches have become the focus of attention, especially applications built using such systems as IBM's *Watson*; and it was clear, on the basis of feedback from our students, that they would like to gain some experience with such applications. As it turns out, there is a fairly large IBM office in central Ohio, it has a fair sized Watson-focused group, and some of our AI faculty had some technical contact with the people in that group. Hence, with some input from the IBM group, we were able to tweak the project in CSE 5914 to be one that builds an interesting application that exploits key machine-learning facilities offered by Watson. Further, we were able to recruit some senior members from the Watson-group as senior adjunct faculty to *teach* some offerings of the course. This version of 5914 has proven quite popular.

On a different note, as is the case with many CS programs across the country, the demand for our BS-CSE program has been steadily climbing. This is matched by the trend in the job market with our recent graduates being highly recruited for interesting and challenging positions. While these are certainly welcome and reflect very positively on the potential of our program and the field as a whole, the increase in demand for the program has put substantial pressure on enrollments in both required and elective courses and we have been working to address this in various ways. First, we have recruited a number of tenure-track faculty, specializing in areas that are likely to be important in the coming years. Second, we continue to look for and hire part-time faculty from among local computing professionals to teach specific courses including some of the capstone design courses where their industry experiences and insight can be of great value. Third, we continue to look for qualified computing professionals with excellent teaching skills who are interested in teaching full-time, to hire them as non-tenure-track faculty. One particularly valuable pool of such candidates is our own recent PhD's, in particular those who have already obtained classroom experience as *graduate teaching associates*. While these efforts have helped, the demand is so great that in spite of these efforts, enrollment pressures in our courses has continued to increase. Hence, a couple of years ago, we instituted a GPA-based enrollment management process to restrict the number of students who are admitted to the major program. This means that a pre-major will be admitted to the major only if, after completing the specified pre-requisite courses (including CSE 2221 (Software I), Math 1151 (Calculus I), Physics 1250, Engineering 1181, 1182, and English 1110), he/she has a specified minimum cumulative-point-hour-ratio (CPHR); currently, this minimum is 3.2 (on a 4.0 scale). This effectively means that well over 50% of otherwise qualified pre-majors, i.e., ones who have satisfactorily completed all the prerequisite courses, will nevertheless not be admitted to the major program; and will have to pursue alternate majors. The CSE Advising Office and the College of Engineering do assist these students in finding suitable alternatives.

Clearly, though, this is not a desirable situation since *computer science & engineering* was these students' first choice. We hope that our efforts at improving our ability to serve a greater number of students pays off in the near future and that we will be able to admit more of these students to the major program.

C. Options

The full details of the BS-CSE curriculum appear under under *Criterion 5. Curriculum*, page 40. Each student's program is required to include courses meeting the requirements of his/her chosen *specialization option*. The possible options are *artificial intelligence*, *computer graphics and game design*, *database systems and data analytics*, *information and computation assurance*, *computer networking*, *computer systems*, *software engineering*, and the *individualized* option. For example, the computer networking option requires the student to have taken the foundational networking course as well as two additional courses from a list of specified courses on network programming, wireless networking, security projects, and network security. The database systems and data analytics option requires the student to have taken the introduction to database systems course, the advanced database systems course, and either the course on data mining or the course on machine learning & statistical pattern recognition; etc. The full details of each option are considered under Criterion 5 but here it is worth noting that the courses needed to meet the requirements of any given option are not *additional* hours added to the student's program; instead, they serve to provide guidance to the student on choosing a suitable combination of technical elective courses, core choice courses, capstone design course, junior project course, etc., that are appropriate, given his or her choice of specialization option, and that meet the other requirements of the program such as the number of credit hours of technical electives.

D. Program Delivery Modes

Conventional day classes.

E. Program Locations

The program is offered primarily on the Columbus campus of The Ohio State University and all graduates of the program come from that campus. The courses comprising the first year of the program that are common to a number of the engineering (pre-)majors are offered at four regional campuses of the university, located in Newark, Marion, Mansfield, and Lima. Additional courses in the BS-CSE program, mainly CSE 2221 and 2231, are offered at the Lima campus and the Marion campus. The CSE faculty in those campuses, Dr. Roman at Lima and Dr. Mirzaei at Marion, work very closely with the course coordinators on the Columbus campus to ensure that the courses that they offer are essentially identical to the ones on the Columbus campus so that when students transfer to the Columbus campus to continue with the rest of their program, they are as well prepared as students who take all their courses on the Columbus campus. All requirements for admission to the major and academic progress are identical on all campuses, and students must complete a campus-change to Columbus to finish their program of study. Students who transition from a regional campus to the College of Engineering must have a 2.5 OSU cumulative grade-point average, a C- or higher in Math 1151, and completed one of the following science courses: Chemistry 1210, Chemistry 1250, or Physics 1250.

F. Public Disclosure

The Program Educational Objectives (PEOs), Student Outcomes (SOs), the annual student enrollment and graduation data are published on the College of Engineering's web site at:

`http://go.osu.edu/CoECSEPEOsAndSOs`

The PEOs and SOs are also published on the CSE Department's web site at:

`http://go.osu.edu/CSEPEOsAndSOs`

They also appear in the undergrad brochure that is on the web site:

`http://go.osu.edu/CSEUndergradBrochure`

Hard-copies of the brochure are available to students and visitors, including prospective students who are considering applying to Ohio State. We should note that the brochure describes not just the BS-CSE program but also the other undergraduate programs offered in the CSE Department.

G. Deficiencies, Weaknesses or Concerns from Previous Evaluation

No deficiencies, weaknesses or concerns were reported following the previous evaluation.

General Criteria

Criterion 1. Students

A. Student Admissions

Admission to The Ohio State University is handled centrally in the *Office of Undergraduate Admissions and First Year Experience* and is competitive. Selection is based on a holistic review consisting of 10 criteria. These criteria are (in no particular order):

- Successful completion of the minimum college prep requirements.
- High school performance(class rank and GPA), including participation in accelerated programs like Honors, AP and IB; Ohio State does not have minimum requirements for class rank or GPA.
- Standardized test scores(ACT or SAT); Ohio State does not have minimum requirements for ACT or SAT scores.
- Ability and desire to contribute to and engage with a diverse campus community.
- Experiences that demonstrate leadership or engaged involvement(e.g., co-curricular activities, work experiences or military service).
- Status as a first-generation college student.
- Demonstration of outstanding talent in a particular area.
- Academic competitiveness of the high school.
- High school performance if adversely affected by physical, mental or learning environment factors.
- Eligibility for and likelihood of benefiting from organized support services at Ohio State.

After admission to the university, an additional review is conducted for admission to the College of Engineering. Admission to the college is competitive and requirements are designed to identify students who are prepared for the rigors of engineering study. Admitted students enroll as *pre-majors*.

There are no stated cut-offs for test scores or grades. Factors considered include:

- ACT/SAT scores with emphasis on math;
- Strong college prep curriculum with emphasis on rigorous course work in math and science;
- Class rank or GPA.

Students admitted to the university but who do not meet the College of Engineering requirements are admitted to the University Exploration program and may apply for admission to the College of Engineering when they meet the following criteria:

- C- or higher in Math 1151 (or equivalent);
- Credit for one of the following science courses: Physics 1250, Chemistry 1210, Chemistry 1250, Biology 2100;
- OSU cumulative GPA of 2.5 or higher;
- At least one term of full-time undergraduate enrollment at Ohio State.

Students admitted to the *regional campuses* may apply for admission to the College of Engineering when they meet the following criteria:

- 2.5 OSU cumulative grade-point average;
- C- or higher in Math 1151;
- Completion of one of the following science courses: Chemistry 1210, Chemistry 1250, Physics 1250.

To be admitted to the BS-CSE major, the following criteria must be met:

- Completion of CSE 2221;
- Completion of Math 1151;
- Completion of Physics 1250;
- Completion of Engineering 1181 and 1182;
- Completion of Engineering 1100.06;
- Completion of English 1110;
- Completion of at least 15 credit hours earned at Ohio State;
- A CPHR (cumulative point hour ratio, i.e., GPA) of at least 3.2 and an MPHR (major point hour ratio) over CSE courses that can be included in the major program of at least 2.0; OR a CPHR of 3.0 to 3.199, completion of CSE 2221, 2231, and 2321, and an MPHR over CSE courses that can be included in the major program of at least 3.2. As explained in the section on *Background Information* (subsection B, Program History), we are using this requirement as a means to ensure that the number of students admitted to the major programs is somewhat in line with our resources, specifically the size of our faculty, available to teach adequate number of sections of the required and elective courses.

During the term in which these requirements are expected to be met, pre-CSE majors apply for admission to the major and are formally admitted to the major at the end of the term.

B. Evaluating Student Performance

Students are advised and their progress monitored through the CSE Advising Office, which is staffed by three full-time professional academic advisors who are assisted by a graduate administrative associate. Students admitted to the CSE major also have an assigned faculty advisor who is available for consultation about curricular, career and other technical matters. The main components of the system for monitoring student performance are as follows:

1. Students are monitored at the end of each term (Autumn, Spring, and Summer). Those students who are not making satisfactory progress are put on *special action probation* and advised of what they need to do to get back in good standing. Failure to overcome the deficiencies (cumulative GPA below 2.0 and/or GPA in CSE courses below 2.0) within an appropriate number of terms –which depends, to an extent, on the individual circumstances of the particular student– may result in dismissal from the program. Probation and dismissal decisions are made by the *Academic Standards and Progress Committee* (ASAP Committee), a standing committee of the College of Engineering consisting of chairs of Undergraduate Studies committees (or representatives thereof) and advising representatives from the various programs. Students who are put on probation receive official notification (via e-mail to the students' official OSU address) from the department that indicates the reasons for the action,

explains what they need to do in order to get back in good standing, and urges them to discuss their situation with the CSE Advising Office. This allows the professional advisors in the CSE Advising Office to suggest possible corrective actions and also to provide referrals to other university resources, as appropriate. A copy of the policies of the ASAP Committee will be available during the site visit.

2. Advising reports recording the progress of their advisees are available to faculty advisors (on request to the CSE Advising Office). The advising report lists all of the courses the student has taken and the grades in each course. Any transfer credits the student has been awarded for courses taken elsewhere are also listed. The report also provides general information about the student such as the high school the student graduated from and the student's test scores (ACT/SAT, AP, etc.). Faculty advisors do not routinely look at their advisees' advising reports; however, in exceptional situations, for example, where a student is considering an unusually challenging technical elective course, having access to the report enables faculty advisors to provide the most appropriate advice based on the individual student's background and abilities. Students can also access their own advising reports through Buckeyelink, Ohio State's online student information system.

3. The Degree Audit is an online program that enables students to verify what requirements they still need to meet in order to complete the program. This enables the students to monitor their progress through the curriculum.

4. With respect to course prerequisites, each course syllabus lists the prerequisites for the course and students are expected to make sure that they have completed all the prerequisites before taking a course. Prerequisites for most core courses are automatically enforced by the Ohio State's student information system, allowing only students who have completed the prerequisites or are enrolled in the prerequisites at the time of course registration to schedule the courses.

   For the courses not automatically enforced, the instructor of each course will typically go over the prerequisites for the course at the start of the term so that students are fully aware of what the prerequisites are and can make sure that they have completed them. In some cases, because of some unusual circumstances, such as part-time employment or internships, etc., a student may not have been able to take a given prerequisite course when it was offered in an earlier term and may, because of his or her particular background and knowledge, feel that he or she can nevertheless succeed in the course in question. In such cases, the instructor talks to the student and, based on the particular student's background and knowledge, may agree that the student may be able, possibly with extra effort, to succeed in the current course. In other cases, a student may contact the CSE Advising Office to see if he or she would be able, because of scheduling issues, to take a particular course without having taken a prerequisite course or possibly take the course concurrently with the prerequisite course. In such cases, the CSE Advising Office refers the student to the course coordinator who again handles the situation in the manner just described. Ultimately, however, it is the responsibility of the individual student to make sure that he or she takes the courses in the prescribed order. Typically, it is rare for a student to take a course without having completed the prerequisites.

C. Transfer Students and Transfer Courses

1. In order to determine if a transfer student is eligiblefor admission to the university, admissions staff review the student's academic record. Additional non-academic factors may also be considered:

- Ability and desire to contribute to and engage with a diverse campus community;

- Experiences that demonstrate leadership or engaged involvement(e.g., co-curricular activities, work experiences or military service);
- Status as a first-generation college student;
- Demonstration of outstanding talent in a particular area;
- High school performance if adversely affected by physical, mental or learning environment factors;
- Eligibility for and likelihood of benefiting from organized support servicesat Ohio State.

2. Once admission to the university is determined, the student's academic record is reviewed for enrollment in the College of Engineering. Transfer students with at least 30 transferable semester hours (or the equivalent), including credit for at least one calculus course, and a GPA of 3.0 or higher (on a 4.0 scale) enroll as pre-majors. Transfer students with fewer than 30 semester hours (or the equivalent) are considered using the freshmen criteria (see Section A).

3. Transfer students who do not meet the College of Engineering direct enrollment requirements begin their studies in University Exploration – Science, Technology and Environment Exploration. Such students may apply for admission to the College of Engineering when they meet the following criteria:

- C- or higher in Math 1151 (or equivalent);
- Credit for one of the following science courses: Physics 1250, Chemistry 1210, Chemistry 1250, Biology 2100;
- OSU cumulative GPA of 2.5 or higher;
- One term of full-timeundergraduate enrollment at Ohio State.

4. Transfer students who are Ohio residents and who have earned at least a 2.0 GPA in previously completed coursework may apply for admission to a regional campus. A review of a student's academic performance in college will determine eligibility for admission. Transfer to the College of Engineering is the same as for transfer students who begin their studies on the Columbus campus in University Exploration – Science, Technology and Environment Exploration.

5. State Mandated Articulation Programs:
The Ohio Transfer Module (OTM), which is a subset or the complete set of a public college's or university's general education requirement that represents a common body of knowledge and academic skills, is comprised of 36-40 semester credit hours of courses in the following fields: English composition and oral communication; mathematics, statistics and logic; arts and humanities; social and behavioral sciences; and natural sciences. Additional elective hours from among the five areas make up the total hours for a completed Ohio Transfer Module. Students are guaranteed the transfer of OTM credits among Ohio's public colleges and universities and equitable treatment in the application of credits to admissions and degree requirements: `https://www.ohiohighered.org/transfer/transfermodule`

The Transfer Assurance Guides (TAGs) are advising tools that include the Ohio Transfer Module, pre-major/beginning major courses called "TAG courses", advising notes, and foreign language requirements, when applicable. All courses in TAGs are guaranteed to transfer and apply directly to the major requirements accordingly. In totality, the TAGs are guaranteed pathways for students and are very useful advising tools for faculty and other advisors. TAG courses are developed, endorsed, and reviewed by faculty in the content areas: `https://www.ohiohighered.org/transfer/tag`

6. For assessing CSE-specific coursework, the CSE department is careful to ensure both that students get credit for coursework they have completed elsewhere and that transfer students are indeed well prepared to continue with the rest of the program. With this in mind, transfer credits are handled through a two-tiered system. The department supplies the university's Transfer Credit Center within the Office of the Registrar with a *transfer-credit course list*, identifying specific courses from specific universities and the equivalent courses in our program. Any student who has taken course X in university Y gets transfer credit for the equivalent course Z of the program as per this list. A designated CSE faculty member, currently Dr. Gojko Babic, acts as the *transfer credit coordinator*. The coordinator works with the Transfer Credit Center to ensure the currency and accuracy of the list. Students also have access to the equivalency list as well as other detailed information about transfer credits on the Registrar's web site.

For courses not listed on the *transfer-credit course list*, students must submit materials, via an online system, for evaluation to the CSE Advising Office. An academic advisor reviews each student's submission to make sure all required materials have been submitted (including the syllabus for each course to be reviewed and the student's initial transfer credit report from the Transfer Credit Center). Once a submission has been verified, the transfer credit coordinator is notified that the materials are ready for evaluation. The coordinator then evaluates each submission carefully; if additional materials are required, the coordinator contacts the CSE Advising Office and an advisor will work with the student to obtain the requested material. If, on the basis of this examination, the coordinator determines that the course covers almost all of the material and at a comparable (or greater) depth as a CSE course, the student is assigned credit for the CSE course. In some cases, the transfer coordinator may not have sufficient expertise in an area to determine if a transferred course is equivalent to a CSE course. In such cases, the coordinator works with the chair of the Undergraduate Studies Committee and the coordinator of the most relevant course to determine the proper course of action. In either case, in arriving at the decision, the main considerations are such questions as whether the text book used in the other university's course is comparable to the one used in our program, whether the list of topics covered and the depth of coverage of the topics are comparable, whether the student obtained a reasonable grade in the course, etc. If, on the basis of this examination, it is determined that the course covers almost all of the material and at a comparable (or greater) depth as a CSE course, the student is assigned credit for the CSE course; the transfer credit coordinator informs the Transfer Credit Center so that the credit can appear in the student's official university record.

One issue having to do with transfer credits concerns the sequence CSE 2221 and 2231. These two courses form a tightly integrated sequence focusing on software engineering issues, rather than mere programming experience in a particular language. We often have students who may have taken one or two courses in programming at another university, and these students often assume that they would get transfer credit for one or both courses in the sequence. But even if the course(s) the student has taken elsewhere did indeed provide him or her considerable programming experience, giving such credit would not be in the best interest of the student unless the student had also learned and internalized the relevant software engineering principles. Hence our policy is that, in general, a student may not receive credit for the first course in this sequence unless the student has taken courses sufficient to allow awarding of credit for *both* courses. In a few cases transfer students may indeed have obtained a sufficient background in software engineering but the programs they come from, as is the case with many programs around the country, do not place the same emphasis on the ideas of design-by-contract that our sequence does. In such cases, the student may be given transfer credit for CSE 2221 and a warning about the expectations and preparation necessary for taking and succeeding in CSE 2231.

D. Advising and Career Guidance

The CSE Advising Office serves as the first stop for any questions that pre-majors and majors may have on all matters related to the program as well as such matters as research and scholarship opportunities, internships, possible full-time positions after graduation, graduate school options, etc.

1. Pre-CSE students are advised by the three professional advisors in the CSE Advising Office. Once they are admitted to the CSE major (as described under “A. Student Admissions”), they are also assigned a faculty advisor. The faculty advisor is usually assigned to the student based on the area of expertise that the student expects to specialize in; and can be changed upon request from the student (to the Advising Office). The faculty-student relationship is expected to be an on-going interaction until the student has graduated from the program, with the faculty advisor providing assistance in areas such as selection of technical elective courses, industry questions, research opportunities, and technical and other content questions. The professional advisors remain available for all program and university concerns, including academic planning, minors, graduation requirements, petitions, and referrals to campus resources. All of this, including information about the assigned faculty advisor, is explained in the letter that is sent to the student (via e-mail) when he or she is admitted to the CSE major. The students’ on-line account (“Buckeyelink”) also includes the name of the student’s current faculty advisor and the name of at least one of the advisors in the CSE Advising Office.

2. Complete information about the program, its curricular requirements, requirements for admission to the major, information about technical elective and specialization options, computing facilities, honors programs, scholarship and undergraduate research opportunities, transfer credits, etc., as well as policies and procedures for graduation, is available to majors and pre-majors from the CSE Advising Office and on the program’s website. The CSE Advising Office also provides, where needed and appropriate, referrals to other services on campus such as counseling and consultation, study skills workshops, tutoring, etc.

3. For new (and relatively new) faculty, detailed information about the advising process and suggestions on how to work with undergraduate advisees is available through consultation with the advising coordinator in the CSE Advising Office (currently Dr Nikki Strader). In addition, an advising workshop for faculty advisors is presented by the chair of the Undergraduate Studies Committee and advising coordinator from the CSE Advising Office on an as-needed basis. This ensures that faculty advisors have a good understanding of the program, its requirements, the various technical elective options and courses, etc., and are able to provide appropriate advice to their advisees. Faculty advisors always have the option of consulting with the CSE Advising Office and/or with the chair of the Undergraduate Studies Committee on questions from their advisees that they are unsure about. Typically, faculty advisors tend to have many questions during their first couple of years and then they become comfortable dealing with most questions on their own.

4. Career guidance is provided by the CSE Advising Office, by faculty advisors, and by the Engineering Career Services (ECS) office. ECS serves CSE students who seek co-op and internship opportunities prior to graduation, as well as those seeking full-time career opportunities following graduation. ECS provides services to students for up to one year after completion of the BS-CSE degree. ECS is heavily utilized: in 2015-16, 88% of the BS graduates (across all the programs that ECS serves, not just BS-CSE) used at least one ECS service in their job searches and 84% of graduates with firm plans at graduation had engineering-related work experience.

E. Work in Lieu of Courses

There is no mechanism for awarding course credit for a student’s work experience. If a student, possibly because of prior work experience, believes that he or she has a sufficiently thorough grasp of the material in a course, the student may contact the course coordinator to arrange to take an examination in which the student's proficiency can be demonstrated. The examination is typically similar to the final examination in the course, but without use of notations or appeal to details that may be specific to the manner in which the course is taught in the CSE Department so that the student is not disadvantaged by being unfamiliar with those specifics; in other words, the examination is intended to test the student's conceptual grasp of the principles as well as the technical details of the topic in question. If the student displays a satisfactory level of performance in this examination, he or she is awarded EM credit for the course. But the situation is rare; no student has requested such course credit in the last several years.

Advanced Placement (AP) credit is awarded based on the requirements set forth by legislation from the State of Ohio and is managed by the Office of the Registrar.

Dual enrollment credit not taken at Ohio State is treated as transfer credit and subject to the process outlined for Transfer Students above. Dual enrollment credit earned at Ohio State is part of the student's Ohio State record and treated as any other Ohio State course.

F. Graduation Requirements

The graduation requirements are summarized in the table.

Graduation Requirements	
	No. of Credit Hours
Engineering Core	20
CSE Core	22
Non-CSE Core	15
CSE Core Choices	20
Math/Sc. Electives	8
Technical Electives	17
General Education	24
Total	126

Name of Degree Awarded: Bachelor of Science in Computer Science and Engineering (BS in CSE).

Graduation Certification: All students complete an online application to graduate at least one semester in advance of their anticipated graduation. Dr. Nikki Strader, the senior advisor in the CSE Advising Office manages graduation. She reviews students' applications, and if the schedule projection the student lists will meet all degree requirements, approves the application in the online system.

Throughout the final semester, the college's graduation coordinator and CSE academic advisors monitor students' academic records to ensure all remaining degree requirements are in progress; e.g., checking that the student does not withdraw from a course that is required for graduation. At the end of the semester, immediately after final grades are posted, the academic advisors update each graduating student's status to show that degree requirements are either complete or incomplete.

The college's graduation coordinator confirms degree requirements are met according to the online degree audit system. Students whose degree requirements have been confirmed as complete by both the CSE academic advisors and the college graduation coordinator have their degrees conferred and receive diplomas at commencement. Students who did not complete requirements receive information about resolving issues and applying to graduate in a future term.

Under exceptional circumstances, students may petition for course substitutions. For CSE courses, the departmental Undergraduate Studies Committee will approve such petitions only if, after careful consideration, it is determined that doing so would not materially diminish the strength of the student's program or compromise the extent to which the student outcomes are achieved. Petitions involving the general education portion or the engineering core of the program are first reviewed by the CSE Advising Office, and then forwarded to the Petitions Committee of the college. Again, these petitions are approved after careful consideration, provided doing so would not materially reduce the strength of the student's program.

G. Transcripts of Recent Graduates

The program is transcribed as "Bachelor of Science in Computer Science and Engineering" on the transcript, with the specialization option (one of Artificial Intelligence, Computer Graphics and Game Design, Database Systems and Data Analytics, Information and Computation Assurance, Computer Networking, Computer Systems, Software Engineering, and Individualized) completed by the student noted as a "subplan."

Criterion 2. Program Educational Objectives

A. Mission Statement

The Ohio State University's vision is to be "the model 21st-century public, land grant, research, urban, community engaged institution". The mission of the College of Engineering is, "[to be] the leader in engineering and architecture education among public universities by creating fundamental and practical knowledge and by preparing professionals ready to sustain and advance our society."

B. Program Educational Objectives (PEOs)

The educational objectives (PEOs) of the BS-CSE program are:

I. Graduates of the program will be employed in the computing profession, and will be engaged in learning, understanding, and applying new ideas and technologies as the field evolves.

II. Graduates with an interest in, and aptitude for, advanced studies in computing will have completed, or be actively pursuing, graduate studies in computing.

III. Graduates will be informed and involved members of their communities, and responsible engineering and computing professionals who take appropriate account, in their professional work, of such issues as privacy, security, copyright etc. in ways that are consistent with the ACM/IEEE Code of Conduct.

These PEOs have been published on the CSE Department's BS-CSE program web site:
`http://go.osu.edu/CSEPEOsAndSOs`
The PEOs are also published on the College of Engineering's web site at:
`http://go.osu.edu/CoECSEPEOsAndSOs`

C. Consistency of PEOs with Mission

The first PEO, in particular that "graduates of the program will be ... engaged in learning ... and applying new ideas and technologies", will ensure that our graduates will contribute to "preparing professionals ready to sustain and advance our society." The second PEO, that (some) graduates will have completed (or be pursuing) advanced degrees in computing, will ensure that these graduates will be "creating fundamental and practical knowledge". The third PEO, that graduates will be responsible professionals and involved members of the communities, will ensure that they will be "community engaged" and that they will mature into "professionals ready to sustain and advance our society."

D. Program Constituencies

The main constituencies of the BS-CSE program are our current students, alumni, and employers of our graduates. One of the key results, over the last several years, from our exit surveys is the very high importance that graduating seniors place on ensuring that our graduates are employed in the computing profession and will be engaged in challenging activities related to recent developments in computing. The first PEO directly addresses this need. Alumni, via results of the alumni survey, have also expressed the importance of ensuring this.

Employers repeatedly stress the importance of ensuring that our students (and graduates) are well prepared to apply the newest ideas and computing technologies to solving a range of problems. Thus the first PEO directly addresses this need. But, unavoidably, given the tremendous pace of developments in the field, the BS-CSE program cannot ensure that its graduates thoroughly understand every recent development and are well-versed in applying every idea, tool and practice. What it can do, and this is addressed by the second PEO of the program, is to ensure that students with the interest in and aptitude for pursuing the most advanced ideas in the field are prepared to pursue and succeed in graduate studies in computing.

Many employers also stress the importance of ensuring that our graduates act in ways that, while meeting their responsibilities to the employer, are also consistent with ethical and professional norms; as well as that they are involved in their communities since, indeed, the welfare and well-being of the community can have a major impact on businesses in the community. Alumni, via results of the alumni survey, have similarly, but to a lesser extent (see below), stressed the importance of our graduates' being responsible CSE professionals. The third PEO of the program directly meets this need.

E. Process for Review of PEOs

We use two processes for the review of PEOs. First is an *alumni survey*, conducted every other year, that is sent to students who graduated from the program two or three years prior to the date of the survey. One of the questions on the survey asks the respondent to rank, on a scale of *Not important, Somewhat important, Important, Very important*, through *Extremely important*, the importance of each PEO for the respondent's professional career. The first PEO, related to the graduates being employed in the computing profession and applying new ideas and technologies was rated, by most respondents, to be "very important" or "extremely important". The third PEO, related to graduates being informed members of their commmunities and responsible professionals, received a somewhat lower rating with most respondents rating it important or very important. The second PEO, related to the graduates pursuing advanced studies in computing, received the lowest rating of the three, the average being "important", although, interestingly, the distribution was bimodal with highs on the lower and higher levels of importance. The survey also asked the respondent to offer ideas for changes to any aspect of the program including the PEOs. Although there were suggestions on changes to specific courses etc., there were none with respect to the PEOs.

The second process for the review of PEOs involves the departmental *Advisory Board*. The board consists of a dozen or so members, many of whom obtained their undergraduate and/or graduate degrees from the department and have then gone on to distinguished professional careers, mostly in the IT industry with a few in academia. The idea is that the members of the board, because of their professional experience combined with their intimate knowledge of the department, will be able to provide deep insight into the needs of our constituents. The board meets once a year near the end of the Spring semester for an (almost) day-long meeting. One part of the meeting is devoted to a presentation about the undergraduate programs in the department including, specifically, the PEOs for the BS-CSE program. In the course of the presentation to the board during its Spring 2016 meeting, there was an extended discussion of the PEOs. During the earlier part of the presentation, one topic that had come up was the importance of ensuring that our students develop an understanding of the ACM/IEEE Code and ability to apply it in practice. The question then in the context of the PEOs was, why not mention the code explicitly in our PEOs? Prior to that point, our third PEO read as follows: "Graduates will be informed and involved members of their communities, and responsible engineering and computing professionals." The suggestion by the board was to try to revise this PEO to include mention of specific concerns such as privacy and include a reference to the ACM/IEEE Code.

Following the board meeting, the Undergraduate Studies Committee (UGSC), a standing committee consisting of a number of full-time faculty closely involved with the program, advisors from the Advising Office, and including student representatives, considered the board's recommendation and, after some discussion, came up with the current version of the PEO; and this was approved by the full faculty during Autumn 2016.

Criterion 3. Student Outcomes

A. Student Outcomes

Students in the BS-CSE program will attain:

a. an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
b. an ability to design and conduct experiments, as well as to analyze and interpret data;
c. an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
d. an ability to function on multi-disciplinary teams;
e. an ability to identify, formulate, and solve engineering problems;
f. an understanding of professional, ethical, legal, security and social issues and responsibilities;
g. an ability to communicate effectively with a range of audiences;
h. an ability to analyze the local and global impact of computing on individuals, organizations, and society;
i. a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
j. a knowledge of contemporary issues;
k. an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
l. an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
m. 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;
n. an ability to apply design and development principles in the construction of software systems of varying complexity.

The student outcomes are published on the CSE Department's web site at:
`http://go.osu.edu/CSEPEOsAndSOs`
They are also published on the College of Engineering's web site at:
`http://go.osu.edu/CoECSEPEOsAndSOs`

Outcomes (a) through (k) correspond closely to the EAC Criterion 3 outcomes but, in some cases, are elaborations of those outcomes. These elaborations are based mainly on the outcomes/characteristics of the CAC Criterion 3; for simplicity, in the rest of this document, we frequently use the term "outcomes" to refer also to "characteristics" of the CAC Criteria. Our outcomes (a) through (k) serve to specialize the EAC outcomes to apply to *computer science and engineering*. Thus, outcomes (a) and (c) above were obtained as a natural combination of the corresponding outcomes in the EAC and CAC Criteria 3. Outcome (b) of the CAC Criterion was substantially different from (b) of the EAC Criterion; hence, we included both in our set; (b) above corresponds to the EAC outcome (b), (l) above corresponds to the CAC outcome (b).

Outcomes (f)–(h) are natural combinations of the corresponding outcomes from the EAC and CAC Criteria. Outcome (k) is a combination of (k) of the EAC Criterion and (i) of the CAC Criterion. Outcomes (m), (n) are outcomes (j), (k) of the CAC Criterion. The remaining outcomes are based directly on the corresponding outcomes in the EAC Criterion. In summary, our outcomes set is essentially a *union* of the EAC Criterion 3 outcomes and the CAC Criterion 3 outcomes.

B. Relationship of Student Outcomes to PEOs

The student outcomes (a) through (n) contribute directly to preparing graduates to attain each of the three PEOs, I, II, and III. Thus outcomes (a) through (c), (e), and (k) through (n) ensure that graduates will be well prepared to succeed in challenging positions in the computing profession thus contributing to achieving PEO (I). Outcomes (d) and (g) will prepare graduates to work effectively as part of teams of CSE professionals, further contributing to their success as CSE professionals. Outcome (i), along with the solid technical background ensured by outcomes (a), (b), (e), and (l) through (n) will prepare graduates to achieve PEO (II), pursuit of advanced/graduate studies in computing. Outcomes (f), (g), (h), and (j) will prepare gradutes to be informed and involved members of their communities and to be responsible engineering and computing professionals, thereby helping them achieve PEO (III).

Table 1. Contribution of Student Outcomes to PEOs

PEO	a	b	c	d	e	f	g	h	i	j	k	l	m	n
I. Grads will employed in computing profession, learning … applying new ideas & technologies …	***	**	***	***	***	**	**	*	**		***	***	***	***
II. Grads with interest/ aptitude will be pursuing advanced studies …	**	**	**	**	***	**	***	**	***		***	**	***	**
III. Grads will be informed/ involved members of communities and responsible professionals …	*	**	***	***	**	***	***	***	***	***	**	*	**	*

Table 1 summarizes the contributions made by the various outcomes to preparing graduates to achieve the three PEOs, with "*", "**", "***" indicting increasing levels of contribution made by the respective outcomes. Thus outcome (a) contributes substantially to PEO I, moderately to PEO II, and minimally to PEO III.

C. Process for Establishment and Revision of the Student Outcomes

The primary process for establishing and reviewing student outcomes consists of the deliberations, in its regular meetings, of the Undergraduate Studies Committee (UGSC). As noted earlier, the committee consists of a number of full-time faculty closely involved with the program, advisors from the Advising Office, and student representatives. Ideas for changes to the outcomes are presented to the departmental faculty, as a whole, for its approval.

A key input for UGSC's process for review of student outcomes comes from the *Exit Survey*, completed by graduating seniors typically during their last semester. One part of the exit survey asks the respondent to rank, for each student outcome, its importance on a scale of *very-unimportant/ somewhat-unimportant/ somewhat-important/ very-important*. The responses are weighted using weights of 0%, 33%, 67%, and 100% to the four possible ratings. Overall, the graduating seniors felt that the outcomes were appropriate with the ones related to computing knowledge and abilities, such as outcomes (a), (c), (k), (l), receiving very high ratings; and ones related to societal issues, such as outcomes (h) and (j), receiving somewhat lower ratings.

D. Enabled Student Outcomes/Characteristics

As described in Section 3.A, our student outcomes, (a) through (n), in effect, is a union of the EAC Criterion 3 (a)-(k) outcomes, the CAC Criterion 3 (a)-(i) characteristics, and the CAC characteristics (j)-(k) applicable to Computer Science programs. The table below shows the correspondence between the CAC characteristics and the student outcomes of the program. The first row of the table lists the CAC characteristics and the second row lists the particular student outcome that corresponds to that characteristic.

CAC characteristic	a	b	c	d	e	f	g	h	i	j	k
Program's student outcome	a	l	c	d	f	g	h	i	k	m	n

All these outcomes are enabled in a range of required courses as shown in Tables 2 and 3. The outcomes listed in the top row of the tables are our student outcomes; the corresponding CAC characteristic is listed (in parentheses) immediately below each student outcome; note that three of our student outcomes do not have a directly corresponding CAC characteristic. They are included in the tables since the information in the tables comes directly from the syllabi, see Appendix A (page 65 onward), for the courses in the first column and those syllabi include information about the contributions made by each course to each of our student outcomes, including the ones that do not have a directly corresponding CAC characteristic.

As in Table 1, "*", "**", "***" in Tables 2 and 3 indicate increasing levels of contribution made by the courses in the first column of the tables to the various program outcomes and, hence, the corresponding CAC characteristics. However, the notation in the current tables has a more specific definition, and is the one used in the syllabi in Appendix A:

- "***" means the substance of the student outcome is a primary theme of the course; a significant fraction of course time (7 hours or more, often woven through the fabric of the course) is directly related to this outcome.
- "**" means the substance of the outcome is a secondary theme of the course; a smaller fraction of course time (3–6 hours) is directly related to this outcome.
- "*" means the substance of the outcome is not a theme of the course, but it is still treated in the course a non-trivial way; a smaller fraction of course time (perhaps 1-2 hours) is directly related to this outcome.

More complete details of several of the courses listed in Tables 2 and 3 are discussed under Criterion 5 (page 40). Here, we note the following. The first set of courses, labeled "CSE Core", consists of two courses on software development, including key ideas related to developing reliable software; two courses on "systems", the first on computer systems including such topics as CPUs, memory structure, etc., the second on operating systems; and two courses on theoretical foundations, the first on discrete structures, the second on algorithms and data structures. These six courses are required of all students and provide a solid foundation in the discipline. As such, as shown in the table, each of these courses contributes to several of our student outcomes and enable the corresponding CAC characteristics.

CSE 2501 is a one-credit hour course on social, ethical, and professional issues in computing that includes oral presentations by students, in addition to student papers, on appropriate topics; Phil 1338 is a 4-credit hour course on the same topic, taught by the Philosophy Department, with the additional course time being devoted to a thorough coverage of ethical theory. Students are required to take one of these courses; if they take Phil 1338, the additional 3 credit hours count towards their *general education* requirements. Thus these courses enable the characteristics related to communication skills, professional and ethical issues, impact of computing on society, etc.

The next four pairs of courses are the *core choice pairs*. Each student is required to take at least one course from each pair; if the student takes both courses of a given pair, the second course counts as a technical elective in that student's program. The courses in each pair are somewhat similar in nature. For example, CSE 3321 of the second pair is on formal languages and automata theory; the other course, CSE 3341, is on programming language concepts. By requiring each student to take one course from each of the four pairs, the program ensures that the student acquires sufficient breadth in more advanced CS topics that build on the six foundational core courses. At the same time, by allowing the student to take both courses in the pair with the second course counting as an elective, students interested in that particular set of topics can go into greater depth. Each of these courses enables a range of characteristics although the extent to which they contribute to specific characterstics varies with the course. Thus, for example, CSE 3341, given the extensive discussion of how programming languages are implemented and the associated project in the course, contributes strongly to characteristic (c); by contrast, CSE 3321, given its theoretical focus, does not contribute as much to enabling this characteristic.

Table 2. Curriculum to Student Outcomes (part 1)

Course	**a** (c.a)	**b** (−)	**c** (c.c)	**d** (c.d)	**e** (−)	**f** (c.e)	**g** (c.f)	**h** (c.g)	**i** (c.h)	**j** (−)	**k** (c.i)	**l** (c.b)	**m** (c.j)	**n** (c.k)
CSE Core Courses														
CSE 2221	***	*	***		**		*		*		***	**	*	***
CSE 2231	***	*	***	**	**		*		*		***	**	*	***
CSE 2321	***		**		*				*		**	**	**	
CSE 2331	***	*	**		**				**		**	**	**	*
CSE 2421	**	**	***		**						***	***	***	**
CSE 2431	**	**	**		***				*		***	**	***	**
CSE 2501/						***	**	***	*	**				
Phil 1338						***	**	***	*	***				
Core Choice Pairs														
CSE 3231	*		**	*	**	*	*	*	*	*	**	**	**	**
CSE 3241	***	*	**	***	**		*		*		***	*	*	*
CSE 3321	***		*		***				**		**	**	*	
CSE 3341	***	**	***	*	***	*	*	*	**	*	***	***	***	***
CSE 3421	***	*	**		**				*	**	*	*	**	*
CSE 3461	**	**	*	**	*	*		*	*	**	**	*	*	*
CSE 3521	***	*	**		**	*		*	*		**	**	*	**
CSE 3541	***	**	**		*	*	*	*	*	*	**	***	***	**
Junior Project Courses														
CSE 3901	**		***	***	**	*	**		*		***	*	*	**
CSE 3902	**		***	***	**	*	**		*		***	**	**	**
CSE 3903	**		***	***	**	*	**		*		***	*	*	**
Capstone Courses														
CSE 5911	***	*	***	***	***	**	***	*	***	**	***	***	***	***
CSE 5912	***	*	***	***	***	**	***	*	***	**	***	***	***	***
CSE 5914	***	*	***	***	***	**	***	*	***	**	***	***	***	***
CSE 5915	***	**	***	***	***	**	***	**	***	**	***	***	***	***

The next set of three courses are the *junior project courses*; the student is required to take one of these courses. Each course culminates in a medium-sized team project, with the different courses focusing on different sets of topics ranging from web-apps to interactive games to system software[2]. Although each course is centered around the team project, there is also a fair amount of instruction devoted to important tools and techniques and corresponding (smaller) programming assignments; this is because most students in the courses, although they would have heard of the particular tools and techniques, would not have much experience with them, including best approaches to using them. Each course also has a strong focus on principles related to effective software development as well as to communication and team skills; thus each enables a wide range of characteristics.

At the same time as the junior project course, or following completion of the course, the student will typically take a number of *technical elective* courses but the particular courses the student takes and hence the corresponding contributions to the various characteristics will vary substantially from one student to the next. Hence, these courses are not included here.

The final set of courses in Table 2 is the set of four *capstone design courses*. As in the case of the set of junior project courses, the student is required to take one of these courses; but unlike in the junior project courses, no significant class time is, typically, devoted to instruction about tools or techniques. Instead, the focus of the course is almost entirely on the large-sized team project. The capstone course serves as the *culminating* piece of the student's program, where the student is expected to apply the knowledge and skills developed throughout the program and work with the rest of his/her team to engage in a significant design and implementation project to meet the needs of the project sponsor. Each of these courses is organized in a similar manner, the primary difference between them being in the specific area that the team project relates to. In addition to the focus on the technical aspects of the projects, each course also focuses on honing the students' team-working skills as well as communication skills. At the end of each semester, each student team from each of the capstone courses from that semester is expected to present its work at a public poster session which also typically includes demo's of the team's (prototype) system. The session is attended by invitees including, especially, professionals from the local IT industry. The teams are also expected to be aware of any ethical and societal issues that their project may raise and present them, as appropriate, in their posters or in answering questions that the visitors to the poster session may have. Thus these courses contribute to each of the program's student outcomes and enable all of the CAC characteristics.

Table 3 shows how various non-CSE courses that students in the program are required to take enable several of the program's student outcomes/CAC characteristics. ECE 2020 is a 3 credit-hour course on analog circuits and gives students a basic introduction to circuits. ECE 2060 is 3 credit hour course on digital logic and develops students' basic intuition about digital circuits. Engr 1181, 1182 are a year-long sequence of two 2 credit-hour courses taken by all engineering (pre-)majors and help lay the foundation for an engineering orientation in the students and contributes to a range of outcomes.

Math 1151, 1172 is a one-year long calculus sequence; 1172 is designed specifically for engineering majors. Math 2568 is a course on linear algebra which serves as an important prerequisite for several of the technical elective courses such as those on parallel computing, machine learning, and neural networks. Math 3345 is a three credit-hour course on foundations of mathematics; it builds on CSE 2321 and solidifies students' understanding of key ideas such as inductive arguments. These courses mainly contribute to enabling the characteristics related to math abilities. Stat 3470 is a standard course on statistics and probability which

[2] CSE 3903, the course on system software, although based on a course we offered for many years under the quarter system, has not been offered since Ohio State switched to semesters in the Summer of 2012. It will be offered for the first time in Autumn '17. Please see Section 4.B on *Continuous Improvement* for an explanation of how we decided to reintroduce a project-oriented systems software course.

provides important concepts that help solidify students' grasp of arguments and results concerning tradeoffs and resource usage in software systems. Phys 1250 is a standard 5 credit-hour course on physics with an extensive lab component that sharpens the basic engineering knowledge and skills of the student; it contributes to enabling a range of related characteristics.

Table 3. Curriculum to Student Outcomes (part 2)

Course	**a** (c.a)	**b** (–)	**c** (c.c)	**d** (c.d)	**e** (–)	**f** (c.e)	**g** (c.f)	**h** (c.g)	**i** (c.h)	**j** (–)	**k** (c.i)	**l** (c.b)	**m** (c.j)	**n** (c.k)
ECE 2020	***	***	*	**	***						***			
ECE 2060	***	***	*	**	***						***			
Engr 1181, 1182	***	**	*	**	***	*	**	*			***			
Math 1151	***	*			*									
Math 1172	***	*			*									
Math 2568	***		*								*			
Math 3345	***				**						*		*	
Stat 3470	***	**	*		*						*			
Phys 1250	***	***		*	*									
Engl 1110							***							
Engr 2367 Second writing*				*		*	***	**	*	**	*			
Social sciences						***	**	**		**				
Historical study						*	**	*		*				
Arts & hum.						*	**	*		**				

English 1110 is a 3 credit-hour course focused on developing students' writing skills and primarily enables student outcome (g) (characteristic (f)). Students are also required to take a *second writing course*; a range of these, in various disciplines, is available but many students take Engr 2367, American Attitudes About Technology, to meet this requirement. Each second writing course not only further develops students' writing abilities, it also contributes to students' understanding of societal issues although the particular topics that the different courses focus on vary from course to course. Thus the second writing course contributes to enabling the CAC characteristics (c), (f), (g), (h), and (i). The last three rows in the table correspond to general education requirements related to social science, historical study, and arts & humanities which we will see in greater detail under Criterion 5.

Criterion 4. Continuous Improvement

A. Student Outcomes

Student Outcomes (SOs) are assessed using a three-pronged approach for assessing the extent to which the various SOs are being attained. Two of the assessments are *direct* assessments, the third is an *indirect* assessment. The details of the direct assessments are considered in A.1 and A.2 below; details of the indirect assessments appear in A.3.

A.1 Program Outcomes Achievement Test (POCAT)

POCAT (Program OutComes Achievement Test) is an *exit test* that all BS-CSE majors are required to take prior to graduation. When a BS-CSE major applies for graduation, generally the semester before the expected date of graduation, he or she is asked to sign up to take POCAT during the next semester. The test is offered once each semester, typically in the fifth or sixth week of the semester. Following the test, refreshments, typically pizza and pop, are served.

Although all CSE students are required to take the test, the performance on the test does *not* affect the grades of individual students in any courses, nor are records retained of how individual students performed on the test. When a group of students takes the POCAT, each student receives a unique code that appears on the student's test but only the individual student knows his or her code. Students are instructed *not* to write their names or other identifying information on their tests. Once the tests have been graded, summary results, organized by this code, are posted on electronic bulletin boards so an interested student can see how well he or she did and how his or her performance compared with that of others who took the test; indeed, if a student forgets his/her code before the results are posted, there is no way to recover it nor any way for anyone to find out how the student performed. This was a deliberate decision since we did not want the students to spend a lot of time preparing for the test. The goal of the test is to help assess the *program*, not the individual students, by assessing the extent to which the students have acquired and internalized the knowledge and skills associated with the various outcomes of the program, not assess the individual students. Initially, there was a concern that if the individual students' performance on the test did not affect them in any tangible way, they would not take the test seriously. Our experience with the test since it was instituted has eliminated that concern; most students seem to actually enjoy taking the test and take it quite seriously. Indeed, students have occasionally been noticed having heated debates about particular questions in the pizza session following the test.

The questions on POCAT are based on topics from a number of required courses, many of the core-choice courses, and the most popular elective high-level courses related to a variety of key topics such as software engineering, formal languages and automata theory, databases, programming languages, operating systems, computer architecture, algorithm analysis, AI, computer graphics, etc. Each question is multiple-choice with two or three questions in some topics and one in others. But they are not necessarily the kind of questions one might find in, say, the final exams of these courses. Instead, they are more conceptual and are designed to test how well students understand key concepts from across the curriculum. Some questions attempt to probe whether a student is able to relate concepts presented in one course to problems and concepts presented in a later, related course. For example, a question may probe whether a student is able to apply the concept of finite state machines (from the course on automata theory) to the problem of designing a tokenizer for a compiler for a programming language.

The ideal POCAT question not only has a specific correct answer but has *distractors* that are so chosen that they correspond to common misconceptions that students might have about the particular concept. It is for this reason that the summary results of a POCAT includes information not only about the percentage of students who answered a given question correctly but also the percentages of students who chose each of the distractors, in other words how many students harbored the particular misconceptions represented by the various distractors about the underlying concept(s). Each question is typically the result of discussions among faculty involved with the corresponding courses. The key goal of the test is to help faculty use the results to identify specific weaknesses in particular courses and help improve the curriculum.

There is one unusual feature of the POCAT questions that is worth remarking on. Each question on the test has, as one of the choices (typically the last one), an answer along the lines of "I don't know". The instructions for the test suggest that the student should pick that answer if he or she has no idea what the correct answer is. Since their performance on the test will have no impact on their record, students who do not know the answer to the question and *know that they do not know*, pick this answer. This means we do not have to worry about the student trying to make guesses and confounding our attempt to pin down misconceptions that he or she may have. Moreover, students choosing this answer deliberately also represents their evolution from being students to becoming CSE professionals following their impending graduation. As students, their main goal tends to be to get the best possible scores in the tests; hence they make even wild guesses if there is a possibility that doing so would improve their scores. As professionals, their goal should be to solve problems; and the first step in successfully doing so is recognizing, where appropriate, that they do not know the answer to some question, so they can seek suitable assistance.

Given the nature of the questions on POCAT, the grading of the tests is essentially mechanical. The faculty members responsible for each question also provide an estimate of the percentage of students they expect to answer the question correctly as well as the particular outcomes that the question is related to. All of this information is included in the summary results that are produced. The final aspect of POCAT is the *evaluation* of the results and arriving at ideas for improvement. The main discussion of the results takes place in the program's Undergraduate Studies Committee (UGSC). The committee consists of several faculty including some who regularly teach the courses included in POCAT; student representatives; and the staff advisor (who also takes care of administering the test). The committee considers such issues as: (a.) Are there any questions for which the percentage of students who got the correct answer differs substantially from the figure that the faculty involved with the corresponding course(s) expected? (b.) Are there any questions for which particular incorrect answers, i.e., distractors that represent particular misconceptions, especially more popular than other incorrect answers? (c.) Are there any longer term trends with respect to questions related to particular concepts? The student members on the committee often provide insight into particular misconceptions that students might have by noting, for example, that a course taught by a particular instructor takes a particular approach to an idea or a topic and that that might lead to certain specific misconceptions with respect, perhaps, to a related concept. And faculty who have taught the particular courses or related courses bring important insights into analyzing and understanding the results.

The student outcomes that POCAT allows us to assess are:

a. an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;

b. an ability to design and conduct experiments, as well as to analyze and interpret data;

c. an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;

e. an ability to identify, formulate, and solve engineering problems;

f. an understanding of professional, ethical, legal, security and social issues and responsibilites;

g. an ability to communicate effectively with a range of audiences;

k. an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;

l. an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;

m. 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;

n. an ability to apply design and development principles in the construction of software systems of varying complexity.

Item (g), ability to communicate effectively with a range of audiences, may seem surprising. How can a test such as POCAT allow us to assess this outcome? The reason it is here is that one of the questions in the test which is based on the junior project course, concerns what issues one should be aware of when creating documenation of a software system and this, of course, relates to (g). Of course, there are other assessments, described later, of this outcome.

The UGSC maintains a website that documents the results of various offerings of the POCAT. The site also includes a (protected) evaluation page that contains a summary of the committee's evaluation of the test results and ideas for possible improvements; and a *question bank* of possible questions for use in future POCATs. Copies of recent POCATs will be available during the site visit.

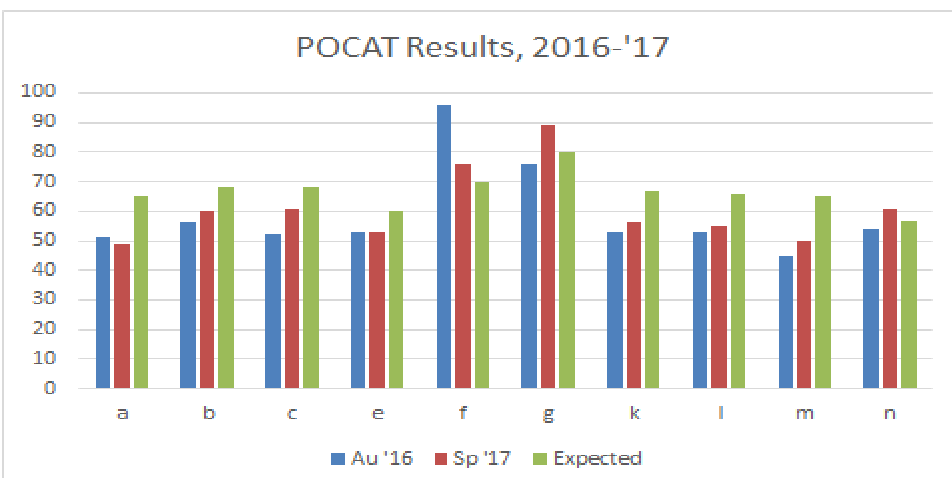


Figure 1: POCAT Summary Results (2016-'17)

Fig. 1 shows the results of the POCAT for Au '16 and Sp '17. One important points worth mentioning here is that, given the broad nature of many of the outcomes, several questions on the POCAT will relate to the same outcome. Thus the results shown in the figure for each outcome was obtained by averaging the student performance or the expected levels of achievement over the questions corresponding to that outcome. The discussion in UGSC and among the broader faculty, however, focuses on individual questions and student performance in them, rather than averages over multiple questions.

A.2 Rubrics

	1	2	3	4
Oral Communication Skills				
a. Organization	Audience cannot understand presentation because of poor organization; introduction is undeveloped or irrelevant; main points and conclusion are unclear;	Audience has difficulty following presentation because of some abrupt jumps; some of the main points are unclear or not sufficiently stressed;	Satisfactory organization; clear introduction; main points are well stated, even if some transitions are somewhat sudden; clear conclusion;	Superb organization; clear introduction; main points well stated and argued, with each leading to the next point of the talk; clear summary and conclusion.
b. Mechanics	Slides seem to have been cut-and pasted together haphazardly at the last minute; numerous mistakes; speaker not always sure what is coming next;	Boring slides; no glaring mistakes but no real effort made into creating truly effective slides;	Generally good set of slides; conveys the main points well;	Very creative slides; carefully thought out to bring out both the main points as well as the subtle issues while keeping the audience interested.
c. Delivery	Mumbles the words, audience members in the back can't hear anything; too many filler words; distracting gestures;	Low voice, occasionally inaudible; some distracting filler words and gestures; articulation mostly but not always clear;	Clear voice, generally effective delivery; minimal distracting gestures, etc., but somewhat monotone;	Natural, confident delivery that does not just convey the message but enhances it; excellent use of volume, pace etc.
d. Relating to audience	Reads most of the presentation from the slides or notes with no eye contact with audience members; seems unaware of audience reactions;	Occasional eye contact with audience but mostly reads the presentation; only brief responses to audience questions;	Generally aware of the audience reactions; maintains good eye contact when speaking and when answering questions;	Keeps the audience engaged throughout the presentation; modifies material on-the-fly based on audience questions and comments; keenly aware of audience reactions.

Figure 2: Rubric for CSE 2501/Phil 1338 (part 1)

The second direct assessment mechanism is a set of three *rubrics* that allow us to assess the extent of achievement of a number of outcomes. We consider each one in turn. The first rubric which appears in Figs. 2 and 3, is for assessing the extent to which CSE 2501/Phil 1338 contribute to communications skills, ethical and professional issues, etc. For each of these, several dimensions are defined by the rubric and, for each dimension, four levels of achievement specified. Fig. 2 depicts the portion related to oral communication skills. Fig. 3 depicts the dimensions related to written skills and to ethical, professional,

	1	2	3	4
Written Communication Skills				
e. Presentation of ideas and organization of the paper	Bland presentation; sequencing and pace of topics seems random; doesn't lead up to any clear conclusions;	Some of the ideas are presented well; others are lacking; offers plausible conclusion(s);	Ideas are well organized and help the reader move along; the key points are presented but does not demonstrate in-depth understanding; leads up to convincing conclusion(s);	The paper is clear and focused; relevant, quality details give the reader important information; helps the reader develop insight into the topic.
f. Style	Occasional problems with word choices and sentence structure, leaving the reader unsure of the meaning; often resorts to jargon/ cliches;	Words and sentences are adequate in general but lack energy; reader has to struggle to keep reading to the end;	Good writing style; sentences flow smoothly and evenly;	Compelling writing style; connects strongly with the reader and keeps him or her engaged right to the end.
Ethical/professional issues, local/global impact, contemporary issues				
g. Understanding of ethical and professional issues	Little or no understanding of professional/ethical issues even where there are serious questions involved;	Some consideration of professional, ethical issues raised directly by the topic under discussion;	Good understanding of and reasonable analysis of all the essential relevant issues.	Deep understanding of the professional issues involved and the ethical implications of the topic under discussion; careful, convincing analysis of all relevant factors.
h. Awareness of implications to society at large	Little or no understanding of (or interest in?) implications to society related to the topic under discussion;	Moderate understanding of the implications to society related to the topic under discussion;	Good understanding of the implications to society of the topic, as well as its relation to general societal issues;	Deep understanding of the immediate and longterm implications to society of the topic under discussion, and the related potential benefits and risks to society.
i. Awareness of contemporary issues (political, cultural, ...)	Little or no understanding of (or interest in?) contemporary issues directly related to the item under discussion;	Moderate understanding of the main relevant contemporary issues directly related to the item;	Good understanding of all the relevant contemporary issues directly related to the topic;	Deep understanding of all the relevant issues, whether political, cultural or other, related to the topic, as well as of issues that may be only tangetially related; good analysis of the issues and possible impacts on various aspects of society.

Figure 3: Rubric for CSE 2501/Phil 1338 (part 2)

and social issues. While a student's abilities related to the other dimensions in this rubric will be evidenced primarily during that student's oral presentation(s) and/or paper(s), the student's abilities with respect to the dimensions in this category are likely to be reflected also and, possibly to a greater extent, in the types of questions that he/she raises during presentations by other students and the types of discussions he/she engages in. For example, if there is a presentation that raises questions related to the security of electronic voting machines, that should present an opportunity, for all students at that session, to engage in a serious discussion about the impact on society of real or perceived insecurity of those machines; similarly for presentations that are related to cyber-espionage; etc. The instructor keeps this in mind when arriving at the assessment of student's abilities with respect to the dimensions in this category.

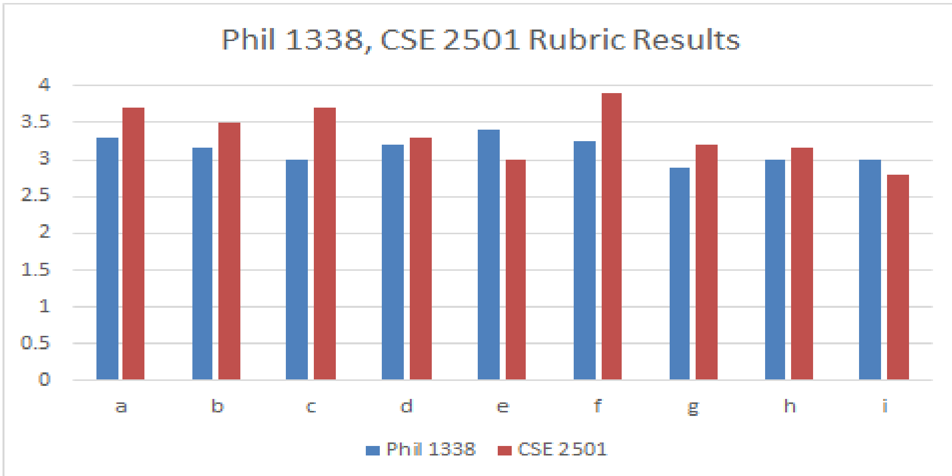


Figure 4: CSE 2501, Phil 1338 Rubric Results (2016-'17)

The results of the evaluation using this rubric appear in Fig. 4 with the labels along the x-axis corresponding to the respective dimensions included in the rubric, see Figs. 2 and 3. These results are presented at a UGSC meeting by the instructors for the two courses and discussed by the committee.

Note: It should be reiterated that the labels along the x-axis of the chart do *not* refer to the *student outcomes*. Indeed, (a) through (f) in the rubric and in the results chart are all dimensions related to student outcome (g), effective communication; and (h) and (i) in the rubric and in the results chart here are dimensions related to student outcomes (f) and (h).

	1	2	3	4
a. Problem formulation	Unclear formulation; Relation to original requirements not mentioned, nor changes in scope	Mostly clear but relation to original requirements and/or rationale for changes in scope not clear.	Satisfactory formulation; Relation to client's original requirements, changes in scope and rationale thereof mostly clear with some gaps.	Excellent problem formulation; Relation to client's original requirements and changes in the scope, if any, explained and justified.
b. Design approach	Poor design; No exploration of alternative approaches; No attention to effective use of resources.	Some attention to alternative design approaches but not a careful analysis of their advantages/ disadvantages; Team picked an approach based on superficial comparisons.	Careful consideration of alternative design approaches and their resource requirements; Not all trade-offs fully analyzed.	Thorough consideration and evaluation of a good set of design approaches; Careful analysis of resource requirements of each and the resulting trade-offs; Where appropriate, client's input sought before making final choice.
c. Implementation (including resource considerations, testing approach, adherence to standards, etc.) If implementation is incomplete, assess based on current state.	Not even basic consideration of memory and other resource requirements; System is very buggy. No systematic testing, nor use of standard approaches/ processes such as agile.	Limited amount of attention to memory and other resource usage; Team has followed a standard (agile/ waterfall/ ...) process but not consistently. Team has put some effort into systematic testing but some bugs remain.	Careful attention to memory and other resource usage and how system might scale with increased demand for services; The team adopted and mostly followed a standard process in its work; The team used a systematic approach to testing and the system seems bug-free.	Meticulous attention to resource usage and to user interface factors; Has ensured that system can evolve to deal with increased demand for services. Team has consistently followed a standard process in its work; Adopted a suitable testing approach, followed it systematically, and thoroughly tested the system. Client involved at all appropriate points.
d. Other factors such as use of professional tools, security considerations, ethical issues.	Little attention paid to factors beyond minimal functional requirements; No systematic use of professional tools; Ethical issues related to system and impact on society not considered.	Some use of common tools seen in earlier courses; Modest effort to ensure basic reliability and security properties; Mostly ignored ethical issues and potential impact on society of systems of this kind.	Good use of professional tools going beyond ones previously seen; System designed to be reliable/ secure under normal operation and under stress; Some consideration of impact of system on society including potential harm system may cause in some situations.	Excellent use of professional tools and systems, identified by careful research; Detailed analysis of security holes with implementation designed to deal with ones that can be reasonably handled and documentation of rest; Analysis of ethical issues related to system and its impact on society including implications of ACM/IEEE Code as it applies to the system, in consultation with client.

Figure 5: Rubric for Assessing Capstone Course Projects (part 1)

The next rubric, which appears in Figs. 5 and 6, is for use by instructors of the capstone design courses to help assess the extent of student achievement of the key student outcomes that these courses contribute to. The first four dimensions, which appear in Fig. 5, address *problem formulation*, *design approach*, *implemen-*

tation, and *other factors* such as use of appropriate tools etc. Again, for each dimension, the rubric specifies four levels of achievement. The last three dimensions (Fig. 6) deal with effectiveness of the teamwork, effectiveness of the team's written documentation, and effectiveness of oral presentations.

	1	2	3	4
e. Effectiveness as a project team	Dysfunctional team; Members blamed each other for problems in project; Team spirit completely lacking.	Team functioned at minimal level of effectiveness; Members concentrated on distinct parts of system without concern for impact on other members' work. In presentations, individual members did not make any attempt to help other members address audience questions.	Generally effective team; Members interested in presenting a positive picture of the team's work; Members helped each other during team presentations. Team members had a general idea of other members' work.	Very effective team; Team members went out of the way to describe how each member contributed to various aspects of project. Team worked as a cohesive unit during presentations, with members seamlessly handing over the conversation from one to another to answer questions, etc.
f. Effectiveness of written communication	Documentation consisted of little more than (poorly commented) system code; Hardly any mention of system's scope, design rationale, implementation choices, etc.	Documentation mostly effective at conveying main aspects of project including scope and design/ implementation choices (but not the rationale behind the choices); Skimpy user manual; Information future teams may need to evolve system lacking.	Team's documentation clearly presented all important aspects of project: original scope, changes made, implementation choices, processes used etc. Test scripts and important parts of code explained; Lessons learned were summarized; Well-written user manual	Excellent documentation; Project's original scope, design choices, relevant code details, processes and tools used, and test scripts all described in a structured and integrated manner; Information to enable future designers to evolve system included; Well-designed user manual provided all necessary information; Illustrations, graphics, and layout executed to excellent effect.
g. Effectiveness of oral communication	Presentations not effective; Failed to present information about some essential aspects of project; Team members ineffective in responding to even simple questions.	Presentations adequate at conveying main ideas behind project including design choices, etc., but not engaging or inspiring. Team responded appropriately to specific questions about specific aspects of project but some responses were unclear.	Presentations were well done and presented all important aspects of project; Team explained rationale behind its choices and summarized important lessons learned; Responses to questions were reasonable although some went into too much technical detail, compromising their effectiveness.	Team's presentations were polished, informative and engaging. In answering questions, the team provided the right level and type of detail for questions ranging from implementation detail to test methodology to future evolution of project.

Figure 6: Rubric for Assessing Capstone Course Projects (part 2)

One important point worth noting is that the effectiveness of oral communication here is, in part, a reflection of the *team's* effectiveness since the presentations were all done by the team as a whole.

The figure shows the results of the use of the rubric for all seven dimensions of the rubric. The rubrics were completed by the instructors of each of the four capstone courses and combined to obtain the results shown in Fig. 7.

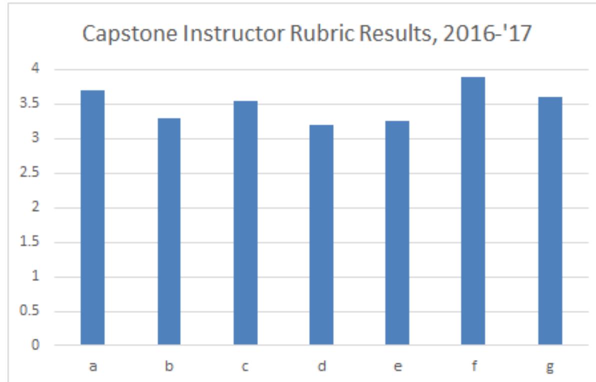


Figure 7: Capstone Instructor Rubric Results

One important activity, as described earlier (page 23), that is part of each capstone course is the *poster session*. Each student team from each of the capstone courses from that semester is expected to present its work at a public poster session which also typically includes demo's of the team's prototype system. The session is attended by invitees including, especially, professionals from the local IT industry. All members of all the teams are expected to attend the session, participate in the demos of the prototype systems and answer questions that the visitors may have.

Our final rubric, which appears on the next page, is for use at the poster session. In designing the rubric, we decided that it would make sense to include essentially the same dimensions as for the instructor with two changes. First, given that the visitors will be looking at the team's poster and interacting with the team members at the same time, it seemed appropriate to combine the two dimensions of that rubric that are related to communication effectiveness into a single dimension for this rubric.

Second, and perhaps more substantive, is that rather than specifying distinct levels of achievement for each dimension, we decided to specify a set of characteristics corresponding to a high-level of achievement for the given dimension and have the visitor determine to what extent he/she agreed that the given team and its poster demonstrated those characteristics. The four possible levels of agreement were converted into the numerical scores of 2 through 5 in arriving at the average scores, shown in Fig. 8, for each dimension for the poster session that was held at the end of Spring '17.

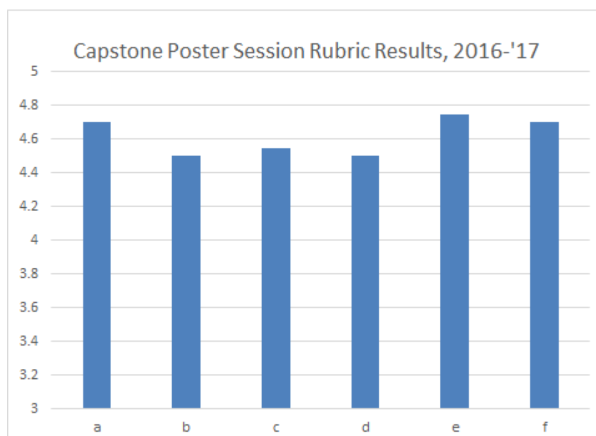


Figure 8: Capstone Instructor Rubric Results

Again, please note that the labels along the x-axes of these charts refer to the dimensions specified in the rubric, not to the student outcomes. Of course, the dimensions in the rubrics correspond to particular outcomes.

Assessment of Poster Presentations in CSE Capstone Design Courses

Code of capstone project being evaluated:

Information about person completing this rubric (check all that apply):

☐ CSE/CIS student ☐ non-CSE/CIS student ☐ CSE faculty member ☐ non-CSE faculty member
☐ IT professional ☐ other (________________)

This rubric and the assessments it provides are an important part of our continuous improvement process designed to help us identify ways to improve our BS program.

The rubric includes six dimensions, these being Problem formulation, Design approach, Implementation approach, Other factors, Effectiveness as a team, and Communication effectiveness, along which the capstone project should be evaluated. For each dimension, there is a description of the corresponding characteristics that are expected of the ideal capstone team and its work. The visitor to the poster session is asked to consider the following statement for each dimension: "Based on what I saw and heard at the poster session, this project team exhibited, in an exemplary manner, all or most of the characteristics corresponding to this dimension"; and then choose one of "Strongly Agree", "Agree", "Disagree", or "Strongly Disagree" (or "Not Applicable" if the item is not relevant to the particular project).

Additional comments related to any of the six dimensions or about other aspects of the project may be entered in the box at the bottom of the rubric.

Problem Formulation	Team has come up with a clear formulation of the problem based on sponsor's goals; any changes in the project scope were clearly explained and justified. ☐ Strongly Agree ☐ Agree ☐ Disagree ☐ Strongly Disagree ☐ Not Applicable
Design Approach	The team has produced a high-quality design that, for the most part, meets the sponsor's goals; in doing so, the team has gone through a suitable iterative process considering various alternatives, including resource (memory, bandwidth, etc.) implications. ☐ Strongly Agree ☐ Agree ☐ Disagree ☐ Strongly Disagree ☐ Not Applicable
Implementation Approach	The team has paid careful attention to all key performance factors that may affect the system. The team has also considered scalability issues as well as possible evolution of the system to meet changing needs. The team has, in its implementation (or plans for it), applied important lessons from key courses in the curriculum; and it has adopted and consistently followed a standard process. **Note: If the team has not fully completed an implementation of the project because of midstream changes in its scope or other reasons, please complete this dimension on the basis of briefly discussing, with the team, its implementation plans and ideas. ☐ Strongly Agree ☐ Agree ☐ Disagree ☐ Strongly Disagree ☐ Not Applicable
Other factors	The team has effectively used appropriate professional tools and systems. It has carefully analyzed its design and implementation to identify potential security holes and documented them. The team has considered the implications of various aspects of the ACM/IEEE Code as it applies to this system and appropriately discussed the relevant questions with the project sponsor. ☐ Strongly Agree ☐ Agree ☐ Disagree ☐ Strongly Disagree ☐ Not Applicable
Effectiveness as a Project Team	The students in this team seem to have worked together very effectively on various aspects of the project from initial formulation based on the sponsor's goals/ requirements, through exploring design alternatives, working on the implementation details, the documentation of the project, through the preparation of the poster. The students also worked effectively as a team in responding to questions and comments from visitors to the poster session. ☐ Strongly Agree ☐ Agree ☐ Disagree ☐ Strongly Disagree ☐ Not Applicable
Communication Effectiveness	The team has produced a well-designed poster that pays careful attention to the items included and the level of detail presented. The poster effectively integrates elements related to basic background of the project with key technical factors. Responses to questions perfectly complemented the poster with the team providing the right level of detail. ☐ Strongly Agree ☐ Agree ☐ Disagree ☐ Strongly Disagree ☐ Not Applicable

Comments:

A.3 Exit Survey

Prior to graduation, BS-CSE majors are required to complete an anonymous *exit survey*. In fact, students complete the exit-survey and take the POCAT in the same session. There are two parts to the survey. The first one asks the respondent, for each student outcome, to rank its importance on a scale of *very-unimportant/somewhat-unimportant/somewhat-important/very-important*; and asks how strongly the respondent agreed with the statement "this student outcome has been achieved for me personally" on a scale of *strongly-disagree/moderately-disagree/slightly-disagree/slightly-agree/moderately-agree/strongly-agree*. In averaging the responses, we attached weights of 0%, 33%, 67%, and 100% to the four possible importance ratings; and weights of 0%, 20%, 40%, 60%, 80%, and 100% to the six possible levels of achievement. The second part of the survey asks students to briefly respond to two questions. The first asks, "What single aspect of the CSE program did you find most helpful? Explain briefly." The second asks, "What single change in the CSE program would you most like to see? Explain briefly." These parts of the survey, although not directly related to specific student outcomes, are naturally very important to students and provide us a good lens through which to view the program and help identify possible improvements.

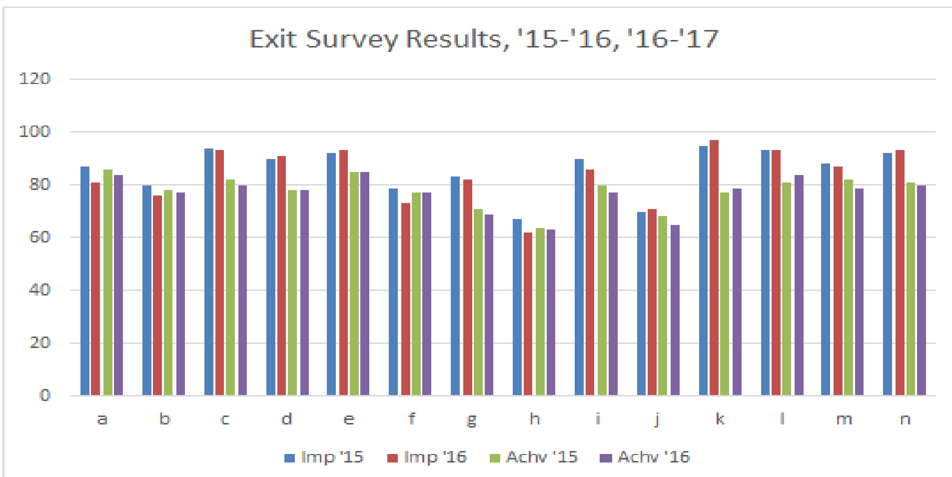


Figure 9: Exit Survey Results (2015-'17)

Fig. 9 shows the results for 2015-'16 and 2016-'17 for the questions concerning the importance and extent of achievement of each of the outcomes.

In summary, our assessment instruments for assessing the extent of achievement of student outcomes consists of: the POCAT; the rubric for use by instructors in CSE 2501/Phil 1338; the rubric for use by instructors in the capstone design courses; the rubric for the capstone poster session; and the Exit Survey. And each assessment is carried out every semester.

B. Continuous Improvement

As already noted at several points, the Undergraduate Studies Committee (UGSC) consisting of CSE faculty, advising staff and student representatives, serves as the main body that coordinates the assessment activities as well as the evaluation of the results. Based on this evaluation, UGSC identifies any problems in the program and possible improvements to address them. The agendas for UGSC meetings are announced in advance to the department's faculty and students. The minutes of the meetings are posted on UGSC's website (and usually include summaries of any extended e-mail discussions that may have occurred after the meeting).

Below we summarize a number of improvements that have resulted from this process. We consider both improvements in a number of individual courses as well improvements in the program. In addition, the process has also helped us improve the assessment instruments; we consider those as well. In some cases, the improvements were based on other information than the assessment results. In each case, we summarize the identified problem, the improvement and the basis for the improvement. In several cases, as we note at various places below, the ideas for the improvement were based on feedback from students rather than directly due to assessment data. In some cases, the improvements are either planned or are still under study.

1. Change in PEO: The second process for the review of PEOs involves the departmental *Advisory Board* (page 18). The board consists of a dozen or so members, many of whom obtained their undergraduate and/or graduate degrees from the department and have then gone on to distinguished professional careers, mostly in the IT industry with a few in academia. The idea is that the members of the board, because of their professional experience combined with their intimate knowledge of the department, will be able to provide deep insight into the needs of our constituents. The board meets once a year near the end of the Spring semester for an (almost) day-long meeting. One part of the meeting is devoted to a presentation about the undergraduate programs in the department including, specifically, the PEOs for the BS-CSE program. In the course of the presentation to the board during its Spring 2016 meeting, there was an extended discussion of the PEOs. During the earlier part of the presentation, one topic that had come up was the importance of ensuring that our students develop an understanding of the ACM/IEEE Code and ability to apply it in practice. The question then in the context of the PEOs was, why not mention the code explicitly in our PEOs? Prior to that point, our third PEO read as follows: "Graduates will be informed and involved members of their communities, and responsible engineering and computing professionals." The suggestion by the board was to try to revise this PEO to include mention of specific concerns such as privacy and include a reference to the ACM/IEEE Code. Following the board meeting, the UGSC, a standing committee consisting of a number of full-time faculty closely involved with the program, advisors from the Advising Office, and including student representatives, considered the board's recommendation and, after some discussion, came up with the current version of the PEO; and this was approved by the full faculty during Au '16. This change is related to student outcome (f).

2. Improvements in required ECE courses: Following the transition to semesters in Summer 2012, the our majors were required to take ECE 2000 and 2100, two 4-credit courses that dealt with analog circuits, digital logic, as well as some material related to signal processing weaving together the material in novel ways. In one of the annual Undergraduate Forums, students expressed serious concerns about the courses. They found the material extremely difficult to understand; moreover, the signal processing material was not only advanced, it was not really directly relevant for most CSE majors. Interestingly, ECE faculty had, at about the same time, started receiving somewhat similar comments

about these courses from ECE majors.

On the basis of this feedback, we worked closely with the ECE faculty and they were able to redesign the courses in a way that would split the two courses into three 3-credit-hour courses, two of which, 2020 and 2060, dealing with analog circuits and digital logic respectively, would be the only ones relevant for CSE majors. Following this change, our program requirements were modified to replace 2000 and 2100 with 2020 and 2060 and adding the additional credit hours to the tech electives of the program. Feedback from the students in more recent forums concerning the new courses has been positive. This change is related to student outcome (c).

3. An IT professional from the central Ohio area who is interested in our program heard about the POCAT and we shared a copy of a recent test with him. He liked the general idea of the test but said that if he were interviewing someone for a position in his company, he would not ask any of the questions that were in the test. This was a serious concern for us because, POCAT being an exit test, our goal was to see, on the basis of the performance of the students on the test, how well they were prepared for the profession. Further discussion with the professional (after he had consulted with some of his industry colleagues) allowed us to pin down the specific issue he was concerned with. It was not that the concepts that the questions in the test were probing were not relevant for software engineers in his company but rather that the *context* in which the questions were presented were too simple, perhaps even simplistic. Of course, trying to set the questions in more realistic contexts would cause difficulties as well because that would require the inclusion of somewhat elaborate explanations in the question and that would change the very nature of the test. But we believe, and the IT professional agrees with us, there might be middle ground possible here and we are exploring ways of reaching that middle ground. This change, once it is implemented, should contribute to outcomes (k) – (n).

4. One of the questions that we have used on the POCAT concerns the relative time it would take two different machines to execute a given program P, given such factors as the cycles per instruction for the compiled version of P on each machine. While this question does probe the student's grasp of important ideas related to architecture, it is easy for the student to pick the wrong numerical answer among the four or five that were listed. Following discussion in the UGSC, the architecutre faculty came up with a more directly conceptual question (concerning pipelining) which we now include in the POCATs. This should allow us to assess the achievement of outcomes (c) and (n).

5. During the capstone poster session at the end of Spring '17, a couple of the visitors to the session noted that the rubric that we had developed (page 34) was not ideally suited for some of the games projects as well as some of the machine learning (Watson-based) projects. They suggested that a key element, i.e., "coolness factor" for lack of a better phrase, ought to included and projects in these genres ought to be evaluated with respect to this factor. We are in the process of trying to come up with language to add to the rubric in such a way that it is somewhat generic but, nevertheless, captures this intent. This change, once implemented, will allow us to evaluate these projects better and, in particular, the extent of achievement of the outcomes corresponding to the dimensions in the poster session rubric.

6. One other observation that was made by several visitors to the capstone poster session at the end of Spring '17 was that there were some easily avoidable flaws in the posters of several of the teams and avoiding those flaws would have substantially improved the posters. In some sections of the capstone courses, the instructors have attempted to talk about useful guidelines to keep in mind when designing the posters but the message doesn't seem to register with the teams, possibly because the discussion

is relatively early in the semester and by the time the teams are working on their posters which is near the very end of the semester, those lessons tend to be somewhat faint memories. We are in the process of creating a written set of guidelines along with online examples of effective posters which will be shared across all sections of the capstone courses. This should improve achievement of outcome (g).

7. Yet another lesson of the capstone poster session at the end of Spring '17 was that it takes much longer than we had expected for a visitor to provide a quality evaluation of a project. Put another way, we had expected each visitor whom we had requested to do the assessments to perform too many of them – roughly 10 in about 60 to 90 minutes; and most visitors found themselves rushing through the projects to get done. We are working on ways to modify the process so that the visitor will have adequate time to spend with each project that he or she will assess (without, however, expecting them to stay much longer at the session); we are shooting to have each visitor evaluate no more than 4–5 projects and have each project assessed by only one visitor rather than two as we tried to do. Again, this change, once implemented, will allow us to evaluate these projects better and, in particular, the extent of achievement of the outcomes corresponding to the dimensions in the poster session rubric.

8. Some of the senior undergraduate students, all members of the ACM-W chapter in our dept., recently undertook, essentially on their own, an initiative to improve student awareness of diversity-related issues and problems by working with the advisors from the Advising Office who teach the 1-credit-hour "survey" class, Engr 1100 (see Table 5-1, page 40). These students had been following the recent stories/scandals surrounding Uber and decided they should do their part to help entering students become alerted to these problems. This is an on-going effort but it is expected that a pilot version should be in place for Au '17; depending on its effectiveness, this will help improve achievement of outcomes (f) and (j).

9. One of the questions that we have included in the POCAT relates to databases concerning the notion of a *key*. This topic is discussed in some depth in CSE 3241, one of the core choice courses that is taken by many students. Unfortunately, performance has been much weaker than is expected by the faculty involved with the course. To pinpoint the precise problem we have tweaked the question in various ways such as, e.g., including language in the question that would remind students of the definition of a *key*; but so far we have not succeeded in pinpointing the problem. We are continuing to work on this problem.

10. Along a different line, and as noted earlier, the *exit survey* includes two questions, one that asks the respondent what single change he/she would like to see made in the program; and a second one that asks what single aspect of the program the respondent found most helpful. One common response to the first question is the addition of more project-oriented courses (such as the course on app development or the one on information security projects). We are indeed interested in doing so and are attempting to hire additional clinical faculty with the right expertise to create and teach such courses. At the same time, several students, in response to the second question, note that while their fellow-students would rather take more project-focused courses, they themselves found the concept-focused courses most useful; these may well be the students who are planning to go on to advanced studies.

11. In one of our annual Undergraduate Forums a couple of years ago, a few students pointed out that there was a duplication of material between CSE 2501 and the (3-cr-hr) Phil 1337 (Ethics in the profession: Introduction to Computing Ethics) which they had taken before 2501 as part of the general education requirement; and the question was, why did they have to take 2501 if they had already taken Phil 1337.

The answer, of course, was that 1337 did not include a strong set of (oral and written) communication activities and that was an important aspect of 2501. The discussion that followed in the forum and in the UGSC meetings after the forum led to our getting in touch with the Philosophy Dept. and, after a very cooperative set of discussions, to the creation of the 4-cr-hr Phil 1338 that effectively combined the content of 1337 and the oral and written communication aspects of CSE 2501 so that now students may take either Phil 1338 or a completely different general education course and CSE 2501; Phil 1337 is still offered but it is intended for students in other majors interested in computing topics rather than CS majors and our students are strongly advised against taking it.

C. Additional Information

All the assessment instruments, including copies of recent POCATs, will be available at the time of the visit. Complete results from all the instruments will be available. Also available will be copies of minutes from UGSC meetings where the results from various assessments were discussed.

Criterion 5. Curriculum

1. Table 5-1

Table 5-1, parts 1, 2, show the four-year plan of study for students in the program, followed by explanatory notes. Most courses are offered every semester; hence the notation "***" is used in the "last 2 offerings" column to indicate that the most recent offerings were in Au '16, Sp '17. The figures in the last column are the maximum section sizes over all sections offered during Au '16 and Sp '17; and the maximum sizes of the recitation/lab sections, if any, indicated in parentheses.

Table 5-1: Curriculum (part 1)

Course (Dept, No., Title)	Reqd / Elective / Selected Elec. (R/E/SE)	Math & Basic Sc.	Cmptng topics Fund. / Adv.	Eng. Topics Check if sig. design	Gen. Educ.	Other	Last 2 ofrngs (year, sem.)	Max. section enrollment last 2 ofrngs (Lecture (Rec.))
Semester 1[1]:								
Engr 1100.06: Engineering Survey	R					1	***	53
Engr 1181: Introduction to Engineering I	R			2 (√)			***	72
Math 1151: Calculus I	R	5					***	520[2] (34)
Phys 1250: Mechanics, Thermal phy., Waves	R	5					***	230 (31)
Semester 2:								
CSE 2221: Software I	R		4(F)	4 (√)			***	44
Engr 1182: Introduction to Engineering II	R			2 (√)			***	72
Math 1172[3]: Engineering Math A	R	5					***	245 (32)
English 1110: First year Engl. Comp.	R				3		***	24
General Education Elective	SE				3		***	varies
Semester 3:								
CSE 2231: Software II	R		4(F)	4 (√)			***	42
CSE 2321: Foundations I	R	1	2(F)	2			***	43
Stats 3470: Probability & Statistics	R	3					***	224
Math/Science Elective[4]	SE	4					***	varies
General Education Elective	SE				3		***	varies
Semester 4:								
CSE 2331: Foundations II	R		3(F)	3 (√)			***	42
CSE 2421: Systems I	R		4(F)	4 (√)			***	42
ECE 2060: Intro to Digital Logic	R		3(F)	3 (√)			***	212(70)
Math 3345: Fnds of Higher Math	R	3					***	33
General Education Elective	SE				3		***	varies

[1]: Students who do not have any computing background also take CSE 1223, *Java Programming, I*, in the first semester. The course is not part of the major program and the hours are not counted in the totals.
[2]: The typical lecture size for Math 1151 sections is 250+ students; but in Sp '17, because of an unexpected situation, two sections had to be combined with the resulting 500+ students in that section.
[3]: Math 1172 is, essentially, Calculus II but is tailored to suit engineering majors.
[4]: The courses in math/science elective group range between 3 and 5 credit hours. Over the course of the program, the student is required to take a minimum of 8 credit hours.

Table 5-1: Curriculum (part 2)

Course (Dept, No., Title)	Reqd / Elective / Selected Elec. (R/E/SE)	Math & Basic Sc.	Cmptng topics Fund. / Adv.	Eng. Topics Check if sig. design	Gen. Educ.	Other	Last 2 ofrngs (year, sem.)	Max. section enrollment last 2 ofrngs (Lecture (Rec.))
Semester 5:								
CSE 2431: Systems II	R		3(A)	3 (√)			***	42
CSE 3901/3902/3903[5]: Jr. Project	R		4(A)	4 (√)			***	33
ECE 2020: Analog Systems & Circuits	R			3 (√)			***	191(59)
Math/Science Elective[4]	SE	4					***	varies
General Education Elective	SE				3		***	varies
Semester 6:								
2501: Soc., Ethical, Prof. Issues in Cmptng. Phil 1338: Introduction to Cmptng. Ethics[6]	R		1(F)	1			***	39
CSE 3231: Software Eng. Techniques[7]/ CSE 3241: Intro to Database Systems	R		3(A)	3 (√)			***	45
CSE 3421: Intro to Computer Arch.[7]/ CSE 3461: Networking & Internet Tech.	R		3(A)	3 (√)			***	40
CSE 3521: AI 1: Basic Techniques[7]/ CSE 3541: Game & Anim. Techniques	R		3(A)	3 (√)			***	44
Math 2568: Linear Algebra	R	3					***	64
General Education Elective	SE				3		***	varies
Semester 7:								
CSE 3321: Automata & Formal Langs.[7]/ CSE 3341: Principles of Prog. Lang.	R		3(A)	3 (√)			***	44
Technical Elective[8]	SE		3(A)	3 (√)			***	44
Technical Elective[8]	SE		3(A)	3 (√)			***	44
Technical Elective[8]	SE		3(A)	3 (√)			***	44
General Education Elective	SE				3		***	varies
Semester 8:								
CSE 5911/5912/5914/5915: Capstone Design Course[9]	R		4(A)	4 (√)			***	26
Technical Elective[8]	SE					3	***	varies
Technical Elective[8]	SE					3	***	varies
Technical Elective[8]	SE					2	***	varies
General Education Elective	SE				3		***	varies
Totals: ABET Basic-Level Requirements[10]		33	53 (F:21; A:32)	60	24	9[8]		

[4]: The courses in math/science elective group range between 3 and 5 credit hours. Over the course of the program, the student is required to take a minimum of 8 credit hours.

[5]: See below for details of the three *junior project* courses. Students are required to take one of these.

[6]: Both CSE 2501 (1 cr. hr.) and Phil 1338 (4 cr. hrs.) are on social, ethical, and professional issues in computing. Phil 1338 also spends time on gemeral ethical theory from a philosophical point of view. If the student takes Phil 1338, the additional 3 cr. hrs. is counted towards general education requirements.

[7]: (CSE 3231, 3241), (3321, 3341), (3421, 3461), and (3521, 3541) are the four *core choice pairs*. The student is required to take one from each pair; and may take the other one as a technical elective.
**Notes continued on next page.

[8]: The student is required to complete 17 credit hours of technical electives. Of these, at least 9 credit hours must be advanced CSE courses; the remaining 8 credit hours may be some combination of CSE and non-CSE courses. Many students take only CSE courses to fulfill these hours; others who may be interested in exploring applications of computing to or interactions between computing and other fields ranging from math to music to biomedical engineering, use these courses to acquire some knowledge of those fields. In the table, we have included all 8 of these hours under the “other” column. We should also note that the different *specialization options* described later in this section all fit in this template thus the information in the table applies to all students.

[9]: There are four different capstone design courses, these being 5911 (Software Applications), 5912 (Game Design and Development), 5914 (Knowledge-Based Systems), and 5915 (Information Systems). Students are required to complete one of these courses.

[10]: The total credit hours of 33, 53, and 60 for math/basic science, computing topics, and engineering topics, each exceed the minimum ABET requirements of 32, 42.66 (CAC requirement for Computer Science programs), and 48 hours respectively.

2. Alignment of curriculum with PEOs

Every one of the CSE courses in the curriculum contributes to developing students' knowledge and understanding of computing concepts and details; a range of courses including the junior project course, the capstone design course, and a number of other project courses that students often choose as part of their technical electives, as well as the various smaller projects in a number of other required and core-choice courses ensure that students skills and abilities with designing software systems applying suitable principles as well as appropriate technologies is well developed. Thus the curriculum contributes solidly toward achieving PEO I, that " graduates of the program will be employed in the computing profession, and will be engaged in learning, understanding, and applying new ideas and technologies as the field evolves".

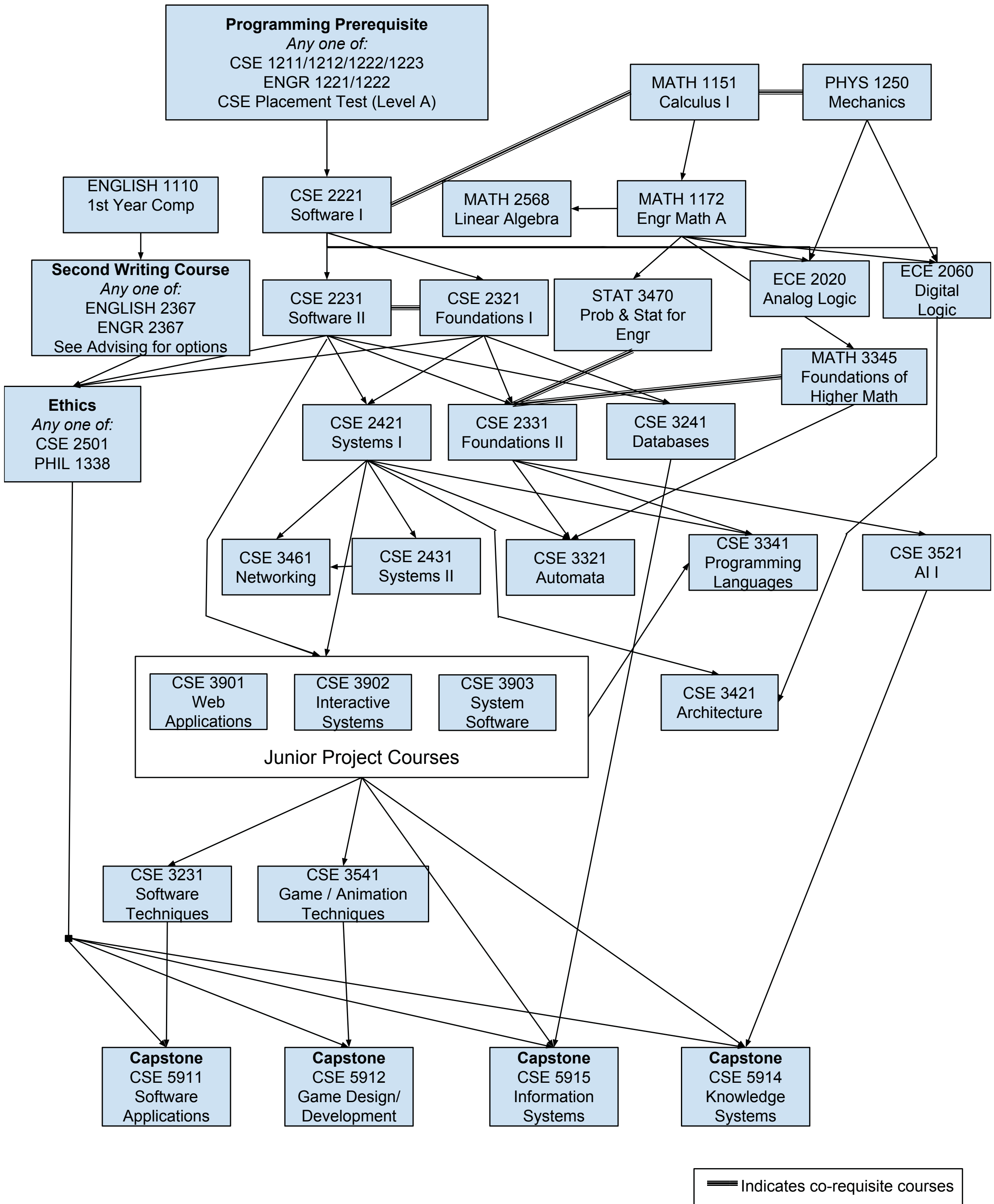
PEO II reads, "graduates with an interest in, and aptitude for, advanced studies in computing will have completed, or be actively pursuing, graduate studies in computing". Math 3345, Foundations of Higher Math, several of the core-choice courses, a variety of advanced courses on such topics as machine learning, network security, computer vision, speech and language processing, cryptography, etc. from which students may choose their technical elective courses help prepare students with an aptitude for graduate studies in computing to be successful in highly-ranked graduate programs. In addition, the option for students to choose 8 hours of technical elective courses from relevant non-CSE disciplines that they may be interested in helps prepare them to pursue those interests in graduate schools.

PEO III reads, "graduates will be informed and involved members of their communities, and responsible engineering and computing professionals who take appropriate account, in their professional work, of such issues as privacy, security, copyright etc. in ways that are consistent with the ACM/IEEE Code of Conduct". CSE 2501/Phil 1338 (both of which include discussion of the Code) and a number of the general education courses, as well as the discussions in some of the CSE tech electives will ensure that students recognize the importance of social and ethical considerations and are prepared to play their roles as informed and involved members of their communities. In addition, in the capstone design courses, students are expected to consider any ethical or social issues that may be related to their projects. Thus the curriculum helps prepare students to achieve this PEO.

3. Curriculum's Support for Attainment of Student Outcomes

Tables 2 and 3 on pages 22 and 24 summarize the contributions made by the various courses to the attainment of the student outcomes and the associated discussion in that section elaborates on the contributions. We will not repeat them here.

4. Prerequisite Flowchart for Required Courses



5. EAC Criteria Requirements for Hours and Depth of Study

The EAC Criteria require 32 credit hours of college level mathematics including probability and statistics and mathematics through differential and integral calculus; and basic sciences, some with experimental experience. The program requires 33 credit hours of college-level mathematics and basic sciences. This includes Stat 3470 (Probabilty and statistics) and Math 1172 which includes material through integral calculus. The 33 credit hours also include Physics 1250 which includes a lab component.

The criteria require 48 credit hours of engineering topics. The program requires 60 credit hours of coures that include engineering topics. The criteria applicable to CSE programs require these topics to include material necessary to analyze and design complex electrical and electronic devices, software, and systems containing hardware and software components. ECE 2020 (analog circuits) and 2060 (digital logic) each requires Math 1172 and Phys 1250 as prerequisites. CSE 2331 (Foundations II: data structures and algorithms) has Stat 3470 as a corequisite which, in turn, requires Math 1172 as a prerequisite. Further, the criteria applicable to CSE programs require discrete mathematics to be included. This is satisfied by the material in CSE 2321 (Foundations I: discrete structures) and in Math 3345 (Foundations of higher math). Thus all of these requirements are satisfied.

6. CAC Criteria Requirements for Hours and Depth of Study

For computer science programs, the CAC Criteria require 42.66 credit hours (one and one-third years) of computer science topics. The program requires 53 credit hours of computer science topics. The criteria further require coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages, and computer organization and architecture. CSE 2321 and 2331 provide coverage of fundamentals of algorithms and data structures. CSE 2221 and 2231 provide coverage of software design and concepts of programming languages. CSE 2421 provides additional coverage of programming language concepts and provides coverage of computer organization and architecture.

The criteria require exposure to a variety of programming languages and systems and proficiency in at least one higher-level language. Students learn to use Java effectively in CSE 2221 and 2231. They learn to use C in CSE 2421 which they also use in CSE 2431. Students are exposed to a variety of computing systems in CSE 2421 and 2431; and, depending on which of CSE 3421 or 3461 they choose, they study the details of advanced architectures or network systems, etc. Thus this requirement is satisfied.

The CAC Criteria requires advanced coursework that builds on fundamental coursework to provide depth. CSE 2431 builds on the systems-foundation provided by CSE 2421. Each of the core-choice courses builds on the content of several of 2000-level CSE courses. Further, the junior project course, the capstone design course, and the various CSE technical elective courses further build on these topics and enable students to apply the ideas in specific software projects that are part of these courses. Thus this requirement is satisfied.

7. Major Design Experience

Four specific courses[3] have been designated as capstone design courses:

- CSE 5911: Capstone Design: Software Applications
- CSE 5912: Capstone Design: Game Design and Development
- CSE 5914: Capstone Design: Knowledge-Based Systems
- CSE 5915: Capstone Design: Information Systems

Depending on his or her specific area of interest, each BS-CSE major chooses one of these courses to meet the capstone design course requirement of the program. The designation of a specific course as a capstone design course is made by the Undergraduate Studies Committee (UGSC) after careful consideration of the course to see how well it meets the intent of capstone design courses. Specifically, the course should be a senior-level course; it should include, as prerequisites, not only the relevant courses in the particular technical domain of the course but also the junior project course and CSE 2501/Phil 1338 since these courses help students develop important professional skills including team-work, communication and lifelong learning, and these skills are key to success in capstone design projects. Design must be the main component of the course and student teams should explore and evaluate possible design alternatives. Where appropriate, consideration of relevant standards must be included; and, again as appropriate, industry standard techniques such as UML should be used in describing designs. Realistic constraints involving, for example, performance (space and time) considerations in the implementation, or platform restrictions imposed by the sponsor/user should be considered. Maintainability should factor into the design; for example, how to ensure that the system can accommodate changing requirements or scale up to meet increased demand, etc. Issues relating to such matters as security, privacy, etc., are occasionally related to the area of the capstone course and students should be consciously aware of these issues, and account for them as needed. Finally, teams should be alert to new methodologies, languages, tools and systems that may be used in industry and that may be relevant to the particular project.

The faculty coordinators for the four capstone design courses, after extensive discussions, have agreed upon a common set of outcomes for these courses. The complete set of outcomes for any given course will consist of these common outcomes plus a set of outcomes that are appropriate to the particular technical domain that a given course belongs to. The common capstone design course outcomes are:

- Master synthesizing and applying prior knowledge to designing and implementing solutions to open-ended computational problems while considering multiple realistic constraints.
- Master deadline driven software design and development in a team setting for an open-ended problem.
- Be competent in evaluating design alternatives.
- Be competent with issues of teamwork, project scheduling, individual and group time management.
- Be competent with presenting work to an audience of peers.
- Be competent with techniques for effective oral and written communication for a range of purposes.

We consider each of the four listed courses and explain how it meets Criterion 5 requirements for the major design experience.

[3] There is a fifth course, CSE 5913, Computer Animation, that was intended to be another capstone design course. However, because of faculty retirement, we have not been able to offer it since 2012 when Ohio State switched to the semester calendar. Hence we do not discuss it here.

CSE 5911: Capstone Design: Software Applications: The prerequisites for this course are the junior project course, CSE 2501/Phil 1338, and CSE 3231, Software Engineering Techniques. The junior project course provides students with a serious experience in software design and implementation, working in teams, and in producing documentation. CSE 3231 provides additional knowledge of software development practices, methods for implementation and maintainence of software, as well as the importance of reliability of software and ways of achieving it. CSE 2501/Phil 1338 introduces students to the ethical issues in computing and provides practice in developing communication skills. All of this knowledge is called upon in CSE 5911.

The main objective of this course is to prepare students for the software engineering profession. Upon completion of the course, students are expected to have applied their software engineering knowledge to aspects of:

- Software design: driven by requirements; scalability, security, usability, and performance considerations; generating and evaluating design alternatives.
- Development process: Configuration management, project management, team structure, roles and responsibities.
- Business issues.
- Presenting work to an audience of peers.

Industry-standard tools and techniques are used extensively in the projects.

Students are organized into teams of 4 to 5 students and work jointly on all phases of the project. The projects have been varied, and are mostly provided by local IT organizations. The external organizations have assigned one or more of their personnel to work with the students – mostly in the provision of requirements, but also as mentors. Industry representatives are invited to the mid-term and final presentations made by the teams, as well as to the final poster session to provide appropriate feedback.

CSE 5912: Capstone Design: Game Design and Development: The game industry is a multi-billion dollar industry, with many complex and challenging computational and user-interaction problems. Many students are interested in working in this field. Many others just enjoy working on projects which provide a challenging set of design and programming problems. This capstone design course provides the opportunities for students to work on a challenging and fast paced project.

The prerequisites for this course are the junior project course, CSE 2501/Phil 1338, and CSE 3541, Computer Game and Animation Techniques. The course requires students to apply the knowledge and skills gained in these courses to develop fully functional computer games in a team-based design and implementation project. The junior project course not only develops key technical skills needed in working on large software projects but also soft skills such as team-working and communication. CSE 2501/Phil 1338 develops oral and written communication skills and equips students with an understanding of social, professional and ethical issues. CSE 3541 provides the essential foundation in computer graphics including in the use of standard tools that are used widely in industry. CSE 5912 specializes all of this to the problem of video game production and introduces additional industry-standard toolkits for graphics, for sound, for game physics, as well as additional open source toolkits for character animation, AI and input controllers.

Students have developed a number of games in different genres over the past several years. Examples include a marble maze, a car simulator, ninja battles, stealth games and RPG games. Students are organized into teams, typically 4 to 5 students per team. The course is organized to push these teams towards a complete game. An Agile development approach is employed with new iterations of the game due every two to three weeks. For the first iteration, a skeletal menu system is developed for starting the game, ending the game, selecting options and playing the game. A splash screen is created to allow the students to highlight the game and themselves. This allows the teams to get organized, develop and refine their ideas for a game while still making progress towards completion. For the second iteration, the students are asked to focus on core components (e.g., resource management, key interfaces, event handling or message passing, etc.) of the game development and basic user interaction. The graphics should be minimal at this stage using boxes and spheres. During this stage the students start to take over the objectives for the following iterations. In total there are five iterations or drops of the game during the course.

Each team is required to make a series of oral presentations spread out over the term; the presentations are devoted respectively to a preliminary proposal for the game, and successive iterations of the game. Students are asked to not only present results, but discuss technical challenges that they are most concerned about that might impact successful completion, and the scope of work for the next iteration. Individual team members may specialize on different aspects of the game. One student typically coordinates the inclusion of sound and sound effects in the game, another may handle all of the physics and another handle any artificial intelligence. Some students may migrate to more of a leadership role, a quality assurance/testing role or a scrum master; etc.

In addition to the Agile method, students use suitable version control systems and develop a web site to highlight their work and keep track of their progress. The demands on the student's time and effort are considerable. However by the end of the course, students have gained a comprehensive experience in design and implementation of video games.

CSE 5914: Capstone Design: Knowledge-Based Systems:[4] In recent years, the use of knowledge based systems and consumer/client facing artificial intelligence have become more common in industry. Products such as Apple's *Siri*, Microsoft's *Cortana*, and IBM's *Watson* have demonstrated concrete applications of machine learning, cognitive computing, and natural language processing. In this course, student play the role of a software startup making use of the *IBM Watson Cloud Services* to design, develop, and present a cognitive-powered application. Some examples of applications developed by students include:

- A personal trainer application, which guides the user through exercise routines and helps plan workout regimens;
- An interactive kitchen assistant, which helps you choose recipes and assists in their preparation;
- A movie recommendation engine, which identifies films similar to those you have liked in the past and seamlessly combines this with your request;
- An voice-controlled quadcopter drone, which interactively answers questions about its own capabilities and responds to detailed voice commands;
- A cognitive job-matcher, which links resumes to relevant job postings;

[4] As noted earlier in the self-study, this course has been revised recently in view of recent developments in the field. Previously, the main project in this course was based on *expert systems*. But, over the last several years, *machine learning*-based approaches have become the focus of attention. Because of this and based on feedback from our students, the faculty involved with the course revised the project in the course to be one that builds an interesting application that exploits key machine-learning facilities offered by tools such as IBM's Watson. The revised course has now been taught several times including, on several occasions, by adjunct faculty from the Watson group at the local IBM center, and has proven very effective as a capstone design course.

- A hostage-negotiator training simulator, in which the client interacts verbally with a simulated hostage-taker.

This course requires students to make use of a variety of software tools and methods. Because of the widely divergent areas of expertise required (UI, cloud computing, machine learning training/maintainance, natural language processing, to name a few), teamwork and communication are essential for success in the class. The class is organized into a set of 3-week *time boxes*, which consist of discrete units of work that are planned and implemented by the teams themselves in an agile framework. At the end of each time box, the teams present to their peers and a simulated client a complete, working product, which is iteratively improved and expanded over the course of the semester. Students also gain experience with a variety of technical topics in parallel study groups, which have historically included topics like Docker, Ansible, build tools like Maven and Gradle, UI Testing with products like Selenium, protocol tools like Thrift and Protocol Buffers, and other topics that are relevant to software development. Teams also present their final results at the poster session at the end of the semester where their projects often attract a lot of attention from visitors to the session.

CSE 5915: Capstone Design: Information Systems: CSE 5915 is a capstone design course focusing on information systems projects. The course involves project design, planning and implementation of the project as a team, and presentation of the project results. It is appropriate for BS-CSE majors planning careers in large IT companies including insurance and banking companies.

The course specifically deals with database design projects. The design activity involves designing the database itself, designing some appropriate indexing and query processing techniques for the particular application, and the design of the interface between the user, database engine and back-end server. Besides the problem solving associated with the design of the project itself and its implementation, three major features of the course, as in the case of all capstone design courses, are documentation, presentation, and teamwork. Documentation tasks required for each project include an initial project proposal, weekly group reports, final individual reports, final group reports, and the development and maintenance of a group web-site. Project teams typically consist of 4 to 6 students. The students typically organize their own teams and select the project they wish to work on. Where needed, the instructor assigns the student to a group based on the student's background and interests. As part of the course, each student is required to make at least one oral presentation. The instructor and class members provide feedback and ideas to the groups following weekly group presentations. Midterm and final demonstrations provide a forum for each group to showcase the design and results of their projects.

The course helps students to improve their skills in individual and group time management, project scheduling, professionalism, communication, and teamwork. Success of the projects is highly dependent on the effectiveness of the teamwork. Students work within teams and develop interpersonal and time management skills while participating in the project design, implementation, documentation, and presentation. Many of the projects in the course are interdisciplinary in nature, and effective utilization of each team member's talents is critical for delivering a good project. Some example projects are:

- Genome Databases: Storing, indexing, and retrieval of large scale biosequence databases;
- Web-based medical information systems: A database system and a user-friendly interface for an online community of doctors and patients;
- News articles website: Indexing and dynamically updating large number of news articles using techniques such as latent semantic indexing.

Groups are required to meet at least twice a week, but typically they meet more frequently.

The course begins with lectures describing the requirements of projects in general and technical details about specific potential projects. After projects are selected, groups provide presentations to the class, which promotes discussion and feedback, and generates ideas about potential improvements. Assignments for the students over the term include a project proposal, weekly progress reports, weekly group presentations, midterm and final demonstrations, individual and group final reports, and the poster presentation at the common capstone poster session.

Summary: The capstone design courses are all well-received by the students. Students have reported that the courses have helped them significantly in their job interviews. Industry sponsors have also generally been pleased with what the students have delivered.

Thus, each of these courses meets the Criterion 5 requirements for the major design experience: it is a culminating activity; it builds on knowledge and skills acquired in earlier course work; and incorporates appropriate standards and constraints in the design/implementation.

8: Cooperative education

The program does not allow cooperative education to satisfy curricular requirements.

9: Materials for review during site visit

Course syllabi, textbooks, and a range of samples of student work for each required course, each capstone design course, and all of the popular electives will be available for review during the site visit.

B. Course Syllabi

Appendix A, starting on page 65, includes the syllabi for all courses used to satisfy the math, science, and discipline-specific requirements.

Criterion 6. Faculty

A. Faculty Qualifications

The tenure-track and clinical-track faculty members of the CSE Department have a broad range of technical interests and expertise. Particular strength in the department exists in the areas of artificial intelligence, computer graphics, networking, security, high-end computing systems, and software engineering, etc. All of the tenure-track faculty and clinical-track faculty have the Ph.D. in computer science or a closely related subject. They typically are quite active professionally.

The department's full-time lecturers and senior lecturers have graduate education in computer science, and in some cases the Ph.D. in computer science. They interact regularly with tenure-track faculty on their instructional assignments, and in some cases also participate in research activities. They also are supported in professional activities including attendance at professional meetings.

Part-time lecturers and senior lecturers also have professional experience equivalent to graduate education in computer science, and most have graduate degrees in computer science. They are practicing computing professionals in the Central Ohio community, and bring a practical side to the delivery of the curriculum. This enriches the students' experience, and students often comment positively about this in the exit surveys.

Table 6-1, Faculty Qualifications, appears on the next three pages. Faculty resumes appear in Appendix B.

B. Faculty Workload

Table 6-2 appears on the next four pages. The order or entries in the table is as follows: tenure-track faculty, followed by clinical-track faculty, followed by full-time lecturers and senior lecturers, followed by part-time (adjunct) lecturers and senior lecturers.

Tenure-track faculty are primarily expected to focus on research including graduate student, especially PhD, advising and in professional activities such as participating in professional conferences; hence their teaching responsibilities are correspondingly lower. In addition, some of the senior tenure-track faculty have major administrative responsibilities (such as serving as chair of major committees) and receive credit for that work as well. Clinical-track faculty are expected to engage to a somewhat lesser extent in research; they are expected to engage to an extent in advising graduate students at the MS level; and in engaging with local industry or participating in other service activities. Full-time lecturers and senior lecturers are expected to focus primarily on teaching of undergraduate courses and engage, to some extent, in professional activities; a handful of them also take an active interest in deliberations concerning the undergraduate programs as well as curricular issues.

C. Faculty Size

The demand for the CSE program has seen rather big swings with the last several years seeing rapid increases. As mentioned earlier in the self-study (see page 6), this has put substantial pressure on enrollments in both required and elective courses and we have been working to address this in various ways. First, we have recruited a number of tenure-track faculty, specializing in areas that are likely to be important in the coming years. Second, we continue to look for and hire part-time faculty from among local computing professionals to teach specific courses including some of the capstone design courses where their industry experiences and insight can be of great value. Third, we continue to look for qualified computing professionals with excellent teaching skills who are interested in teaching full-time, to hire them as non-tenure-track faculty. While these efforts have helped, the demand is so great that in spite of these efforts, enrollment presssures in our courses has continued to increase. Hence, a couple of years ago, we instituted a GPA-based enrollment management process to restrict the number of students who are admitted to the major program. This means that a pre-major will be admitted to the major only if, after completing the specified pre-requisite courses, he/she has a specified minimum cumulative-point-hour-ratio (CPHR); currently, this minimum is 3.2 (on a 4.0 scale). This effectively means that well over 50% of otherwise qualified pre-majors, i.e., ones who have satisfactorily completed all the prerequisite courses, will nevertheless not be admitted to the major program; and will have to pursue alternate majors. Clearly, this is not a desirable situation since *computer science & engineering* was these students' first choice. We hope that our efforts at improving our ability to serve a greater number of students pays off in the near future and that we will be able to admit more of these students to the major program.

D. Professional Development

The university supports a faculty professional leave (sabbatical) activity after every seven years of service. Many faculty take advantage of this opportunity. The department has recommended all requests for professional leave in the last five years. The Department Chair also routinely supports, from the department's budget, travel by all full-time faculty (including Lecturers and Senior Lecturers) to professional meetings, including workshops that promote effective teaching.

Each newly hired faculty is offered a generous startup package to cover their first one to three years of expenses for research students, summer salary, travel costs, and others. The startup funds remains in their accounts even after they get their own grants.

E. Authority and Responsibility of Faculty

Two faculty committees have primary responsibility for all instructional programs in the department, the Undergraduate Studies Committee (UGSC) and the Curriculum Committee (CC). These two committees are made up of a cross-section of faculty members, the professional undergraduate advisor, and student representatives. All faculty are welcome to attend the meetings of the two committees. The chair of each committee is also a member of the other committee, enabling the two committees to work closely with each other. UGSC is responsible for the development and implementation of the processes for assessment, evaluation and improvement of the BS-CSE program including its PEOs and SOs. It is also responsible for documenting these activities. Proposals for major changes in the program are discussed by UGSC before being recommended to the faculty as a whole for its approval. CC is responsible for all courses. Proposals for changes in existing courses as well as for new courses are discussed by CC before being recommended to the faculty for its approval.

At the level of the college, there are two committees that are responsible for all programs in the college. The College Committee on Academic Affairs (CCAA) consists of faculty representatives from the various programs in the college. CCAA is responsible for all academic programs in the college. All new course proposals, proposals for (substantive) changes in existing courses, and all proposals for changes in a program's requirement have to be approved by CCAA before they go into effect. An important subcommittee of CCAA is the Academic Standards and Probation (ASAP) Committee which is responsible for monitoring student performance, and probation and dismissal decisions. Each program in the college has a faculty representative on the ASAP; staff advisors are also part of the ASAP subcommittee. The second committee is the Outcomes Assessment Committee which consists of faculty representatives from the various programs in the college. The committee's main purpose is to serve as a vehicle for programs to exchange experiences and ideas directed towards continuous program improvement.

Criterion 7. Facilities

A. Offices, Classrooms and Laboratories

1. All full-time faculty have private or, in the case of some full-time lecturers, semi-private offices with appropriate office equipment. Part-time faculty typically share offices. All faculty members' work areas are equipped with high-end desktop computing facilities including high-speed internet service. Wireless access is available in all faculty offices. Most administrative staff also have private offices with suitable office equipment, including standard desktop computers and including high-speed internet service. Advisors in the Advising Office are also provided a Microsoft Surface tablet. These devices allow the advisors to be effective in various settings, such as freshman orientation, recruitment sessions, etc., involving large number of students or prospective students and parents. The devices also allow the advisors to be effective when making presentations in the Survey class (Engineering 1100) that all freshmen take during their first semester; etc. Graduate teaching associates have desks in larger shared offices and are provided standard desktop computing facilities.

2. Nearly all OSU classrooms are equipped with a desktop computer with LCD display connected to the internet, attached to a projector and a drop-down screen for display to the class. Instructors may also connect their own laptops/tablets to the projectors.

3. Laboratory facilities: The only laboratory facilities are computing facilities which are described next.

B. Computing Resources

The CSE Department's computing facilities are separate from the general university computing labs, and provide access to UNIX and Windows systems in all faculty and staff offices and in all laboratories used by students in the program. Windows 10 and Red Hat Enterprise Linux 6 and 7 are the operating systems currently deployed. The Department maintains computing facilities in 4 buildings for the program that include roughly 130+ seats for open lab use. Most computing labs for general use by students in the program are open weekdays from 8 am to 10 pm, closed on Saturdays, and open on Sundays from noon to 10 pm. One of the labs is accessible to students in the program *24/7* via their *BuckID* (OSU's student id card). There is another (newly renovated) 42-seat *closed* laboratory that is used for the introductory computer science sequence (and for other courses on occasion). Remote access to the centralized (Windows and Linux) servers is available to all students in the program.

CSE provides a unique instructional experience to the future leaders in our field. Modern development tools such as Eclipse, Microsoft Visual Studio 2015 and Xcode are available in all operating system environments. A special graphics lab provides high-end equipment for graphics related course work.

Virtualization is employed to provide self-contained enterprise-like environments for capstone courses through our Instructional Enterprise Support project. Virtualization is used in many courses that require students to have administrative level access that would otherwise not be possible in the more restrictive instructional computing setting. Virtualization has afforded CSE much success in giving courses what they really need to be truly successful and will play an increasing role in our general services strategy to provide robust and flexible solutions to the Department.

The CSE ethernet network employs redundant 40 Gb links within the data center to connect users to high-powered servers for both computer power and file management. Printing facilities exist in each lab (except for the closed lab, whose print facility is a short distance down the hall). Appropriate software resides on the servers and is accessible by all student and faculty accounts from any lab seat or office.

The department controls two classrooms that have internet access and permanent attached digital projectors, including the closed laboratory classroom used by the introductory sequence. Other classrooms assigned to our courses have generally been upgraded by the University to contain podium computers and projection equipment as standard. The department maintains laptops and laptop projectors that may be reserved by faculty and graduate teaching associates for short term use.

The university's *Office of Distance Education and eLearning* (ODEE) is responsible for providing access to general computing facilities for all students at the university. Various computing labs and "collaboration spaces" are maintained by ODEE throughout the campus. See:
`https://odee.osu.edu/public-computing`

C. Guidance

The main open lab is staffed with student "consultants" who offer help with systems and programming questions. CSE also has a Help Desk staffed with student employees, mostly from CSE/CIS majors. The Help Desk is available to answer more complicated systems questions and resolve any hardware or software problems. Detailed information is available about the computing resources and guidance is available on-line to faculty, students and staff at:
`https://cse.osu.edu/computing-services`

D. Maintenance and Upgrading of Facilities

Until two or three years ago, the CSE Dept. had its own computing staff that was responsible for maintaining our facilities. About three years ago, the College of Engineering decided that it would be better to consolidate the computing resources in the different departments in the college in order to minimze duplication and maximize sharing of resources. At the same time, there was concern that "removing local control" from CSE, given the unique computing needs of our students and faculty, may create its own problems. In order to address these conflicting considerations, the newly created *Engineering Technology Services* (ETS) is centered in the college and reports to the college leadership. However, ETS has 9 employees embedded in the CSE Department to maintain the facilities used in the program by CSE students, faculty, and staff. This allows the computing staff to be more responsive to the needs of the program and can plan upgrades and maintenance activities to minimize disruption and accommodating our needs. The CSE Department's *Computer Committee* is responsible for computing policies and plans. The director of the department's computing staff, student representatives, and a number of faculty are part of the committee and help inform the policies/plans as they are developed. The structure of ETS and precise details of allocation of resources for maintaining and upgrading the computing infrastructure in the CSE department are still evolving; the goal is to maintain the quality of the computing resources available to CSE students and faculty while also working toward consolidation with the computing resources of the other departments in the college.

Equipment upgrades generally occur on a 4 to 5 year cycle for desktop and lab equipment. With server virtualization the upgrade cycle is not as critical since additional resources can be allocated to servers on the fly. The VM infrastructure servers themselves are replaced once they can no longer be warrantied. The entire server hardware infrastructure is currently in the process of being upgraded with the exception of some of the VM host servers.

E. Library Services

The CSE program is served mainly by the *18th Avenue Library* (a.k.a. the Science and Engineering Library). The library is accessible 24/7 to students, faculty, and staff (BuckID) access between 11:30 pm and 7:30 am) with the exception of major holidays (such as Independence Day). Materials may be checked out between 8 am and 11:30pm. Reference librarians are available between 9 am and 5 pm Monday through Friday for assistance with finding books, electronic resources, and other information needs or questions.

The library acquires a variety of printed materials, including textbooks, conference proceedings, journals, and reference materials each year in the area of computer science and engineering. The current collection in CSE is extensive. In addition, students, faculty, and staff may request printed monographic items through OhioLINK, the state of Ohio's interloan library system.

The University Libraries has access to a variety of online resources in the area of CSE. All users have access to full-text resources, such as IEEE Xplore, ACM Digital Library, and Safari Tech Books Online. In addition, full-text access to a large number of journals in computer science is available through OhioLINK's Electronic Journal Center. The list of online journals is available here: `http://go.osu.edu/ListOfOnlineCSJournals`

Other databases, such as the EBSCOhost databases and Lexis Nexis, are also available. Several research databases that abstract or index information in computer science, including MathSciNet, NTIS, Applied Science and Technology Index, Compendex and Compendex Historical, INSPEC and INSPEC Archive, and ISI Citation Databases (included Science Citation Index Expanded) are also available. Faculty, students, and staff may suggest titles that they believe should be included in the library's collection.

F. Overall Comments on Facilities

CSE's computing staff has been extremely effective in maintaining the facilities used by faculty, staff, and students. As long as the budget situation remains stable, the program will continue to provide excellent facilities to students, faculty, and staff.

Criterion 8. Institutional Support

A. Leadership

Professor Xiaodong Zhang is the Chair of the CSE Department. He has been serving as chair since January 2006. He is expected to step down at the end of Summer 2018. An internal search for a new chair is currently on-going and is expected to identify, as the next chair, one of the current full-professors in the department; the search is expected to be complete by the start of or early in Fall 2017. The individual, once identified, will be appointed as "chair-designate" and will work closely with Professor Zhang so that by the time he/she takes over in late 2018 will be ready to "hit the ground running".

The department has a decentralized faculty governance structure. Faculty committees conduct discussions and make recommendations on important issues to the faculty and to the chair, including faculty recruiting, graduate studies, graduate admissions, computing services, curriculum, and all aspects of the undergraduate programs. An *executive committee*, composed of the chairs of the major faculty committees, provides the chair advice on various matters affecting the department and that may not fall within the purview of any particular committee. The chair also organizes occasional staff meetings to discuss administrative issues with appropriate staff members on budget administration, grant administration, graduate admission and support, human resource services, support for faculty recruiting, and various other aspects where they may have special insight.

The department chair has ultimate responsibility for all aspects of the CSE Department, overseeing the budget, and managing the staff and faculty. He is assisted in this task by an *associate chair*, currently Professor Rephael Wenger. The associate chair is responsible for day-to-day matters including, especially, staffing all courses (undergraduate, graduate, and service courses) and for faculty teaching assignments.

B. Program Budget and Financial Support

The University budget model for allocating annual funding to each department is defined mainly by two factors: the total number of credit hours taught by a department, and the total amount of annual faculty research expenditure in the department. Since the CSE Department has maintained a strong levels of both student enrollment and research activities, the budget allocation to the department has been adequate in spite of cuts at the state level. For example, during each of the last several years, the department has been autohorized to hire a number of tenure-track and clinical-track faculty in a number of different focus areas; at the same time, like many other CS departments, we have not always been successful in filling all these positions. Under the budget model, we are also able to hire full-time lecturers and senior lecturers in order to meet the needs of a burgeoning undergraduate student enrollment. Several of the senior lecturers are recent CS PhDs who have a strong interest in pursuing academic careers with a teaching focus; others are professionals with several years of industry experience who have decided to shift gears and focus their careers on university-teaching.

The department has a number of funds mostly endowed by former students and other friends of the department. The income from these funds are used to provide annual scholarships to both undergraduate and graduate students based on merit evaluation, and to support faculty and staff members in career development activities, such as attending conferences on research and education.

Courses at all levels, from service courses through pre-major courses, required and elective courses for majors, and graduate courses, are carefully planned in terms of size, structure, and teaching staff (instructors, TAs and graders). The computing staff works closely with the course coordinators to ensure that any special computing needs of the course are met. All courses taught by faculty have assigned graduate or undergraduate graders who, typically, grade homeworks, lab projects, and the like. Grading of both midterm and final examinations, as well as assigning of final course grades, is the responsibility of the course instructor.

In some cases, advanced graduate students are assigned to teach sections of pre-major and some of the beginning major courses under the supervision of the course coordinator. In the case of GTAs (graduate teaching associates) assigned to teach a major course, they typically go through advance training by a faculty member, often the course coordinator, with considerable experience in teaching the course; this involves sitting through all the lectures, in a semester prior to the one in which the GTA is expected to independently teach the course, of a section of the same course taught by the faculty member and grading that section of the course. Each year, excellent teaching by GTAs is recognized with a certificate and cash awards.

One important university-wide resource that is dedicated to excellence in teaching and that is available to both GTAs and faculty instructors is the *University Center for the Advancement of Teaching* (UCAT) (`http://ucat.osu.edu/`). UCAT's primary goal is to help "Ohio State's teachers approach their work in a scholarly and reflective way, engaging with the research on effective pedagogies, thus promoting continuous improvement of student learning." UCAT not only offers workshops on various aspects of teaching, it also offers individual consultations for interested faculty; these consultations may be followed-up by midterm interviews with students (without the instructor being present) to gather feedback from them; classroom observation where one of the UCAT professionals attends one or more of the instructor's lectures and comes up with specific suggestions on how the instructor might improve his approach to teaching or particular course materials, etc.

The CSE associate chair is a member of the Undergraduate Studies Committee as well as the Curriculum Committee. This ensures that the associate chair is well aware of the overall needs of the program as well as of individual courses including the demand for particular elective courses; and can take these factors into account when planning course offerings. The associate chair also works closely with the Advising Office to ensure that he is aware of any specific concerns or needs that the advisors may identify to help specific individual students or all students in the program as a whole to succeed. This organizational structure, combined with more or less stable resources that have so far been adequate, help BS-CSE majors to succeed in the program and to achieve the program's student outcomes.

C. Staffing

The department currently has 8 full-time administrative staff members in charge of such activities as HR, grant administration, budget, public relations and communications, etc. These staff members help the faculty, the chair, and the associate chair ensure smooth functioning of the department and help address any administrative issues as they arise. Also, as noted under Criterion 7, subsection D, the college's *Engineering Technology Services* unit has 9 technical staff who are embedded in the CSE Department to maintain the facilities used in the program by CSE students, faculty, and staff. In addition to the 8 administrative staff members, the department has an undergraduate Advising Office staffed with three full-time professional advisors and one graduate student assistant; some of the important roles that the staff in the undergraduate Advising Office play in the program is described under Criterion 1, page 4. The strong academic environment, supportive office environment and competitive salary and benefits allows us to recruit and retain highly-qualified and capable staff members who, in turn, provide excellent service to students and faculty.

D. Faculty Hiring and Retention

A key component for hiring of tenure-track and clinical-track faculty is the faculty search committee consisting of faculty representatives in all research areas, and a graduate student representative. The chair is also a member of the search committee. The application files are open to all tenure-track faculty. The reviews of faculty candidates, selections for campus interviews, and final selections are discussed in the faculty search committee, and meetings are open to all faculty.

The associate chair has the primary responsibility for hiring lecturers and senior lecturers. He works closely with all other faculty who may have leads on possible suitable candidates and try to recruit them. As noted earlier, one pool of candidates is our own PhD students who have demonstrated good performance as graduate teaching associates.

Faculty retention is seriously considered in the department. In the periodic merit evaluations, highly productive faculty in both research and high quality teaching are appropriately rewarded with salary increases and "best teacher awards." We also nominate, for various prestigious awards in the college and the university, faculty who exhibit exemplary performance. Over the years, we have been very successful in retaining both tenure-track and non-tenure-track faculty.

E. Support of Faculty Professional Development

Newly hired tenure-track and clinical-track faculty members are offered generous startup packages to cover their first one to three years of expenses for research students, summer salary, travel costs, etc. The department supports regular sabbatical leaves for its faculty. The department also provides travel funds to faculty, including lecturers and senior lecturers, who do not have external funds, to attend at least one professional conference each year.

Program Criteria

The program satisfies the EAC Program Criteria applicable to programs that include the word "computer" in the title and the CAC Program Criteria applicable to computer science programs, as detailed under the Curriculum criterion, see page 45.

Appendix A – Course Syllabi

Syllabi for all CSE courses, as well as mathematics, science, and engineering courses that are required in the program, appear in the pages that follow. The syllabus for each course is in the format specified in the self-study template and includes the items of information listed in the template.

In the syllabi of individual CSE courses, course outcomes are stated in terms of intended student learning outcomes; i.e., each objective is implicitly prefaced by "The student is expected to . . . ". We use the following terminology to describe familiarity level (most to least) with respect to various kinds of material and procedures. A student who receives an "A" in a course should have met substantially all the outcomes as stated, and a student who merely passes the course should have met all or most of the objectives at least at the next lower familiarity level.

- Master means the student will be able to exhibit knowledge of the material and/or skill with the procedure in a new context or novel situation, even when not instructed to do so.
- Be competent with means that the student will be able to exhibit knowledge of the material and/or skill with the procedure in a routine situation such as those covered in the course, even when not instructed to do so.
- Be familiar with means the student will be able to answer questions about the material and/or to use the procedure in a routine situation such as those covered in the course, when instructed to do so.
- Be exposed to means the student will have heard the term and/or seen the procedure, but may not be able to discuss or use it effectively without further instruction.

The syllabus of each course contains a table summarizing the relation between the course and various Student Outcomes of the program. This table contains the same information as Table 2 on page 22 but for just this course. And the notation used is the same as in that table:

- *** means the substance of the Criterion 3 or program outcome is a primary theme of the course; a significant fraction of course time (7 hours or more, often woven through the fabric of the course) is directly related to this criterion or objective.
- ** means the substance of the Criterion 3 or program outcome is a secondary theme of the course; a smaller fraction of course time (3–6 hours) is directly related to this criterion or objective.
- * means the substance of the Criterion 3 or program outcome is not a theme of the course, but it is still treated in the course a non-trivial way; a smaller fraction of course time (perhaps 1-2 hours) is directly related to this criterion or objective.