EVERGREEN VALLEY COLLEGE

Computer Science

Program Review

2015–2016

. . .a place where learning
is everyone's responsibility

Department/Program Name: Computer Science

Last Review: 2011

Current Year: 2015

Preparers' Names: J. Henry Estrada

Area Dean: Michael Highers

Summary of the Department/Program ("Program")

1. Provide a brief summary of your program. Assume the reader does not know anything about it. Your explanation should include a brief history and a discussion of any factors that have been important to the program's development. Please explain the purpose of your program, what students you serve, what services you provide, and why these services are valuable.

The Computer Science department was added to the Division of Business and Workforce Development at Evergreen Valley College in 1990. It was developed at the invitation of UC Berkeley to address the needs of students who lived in our service area and wanted to transfer and major in Computer Science at UC Berkeley. Coincidentally, a new department in Computer Science at San Jose State University was initiated about the same time, and Evergreen Valley College negotiated an articulation agreement for transfer of our Computer Science courses to both institutions. There were originally four courses offered by the department: ComSc 072 (Discrete Mathematics), ComSc 075 (Computer Science I), and ComSc 076 (Computer Science II), and ComSc 081 (Computer Organization).

Qualified faculty from the Computers and Information Technology (CIT) and Math departments taught these courses. EVC offered Discrete Mathematics and Computer Science I in the fall of 1990, and thereafter, all three of our courses were offered every semester from spring of 1990 through the spring of 1996. Beyond its first offering, EVC could never enroll enough students for our course on Computer Organization (principally because the course was, at that time, in the upper division at San Jose State University), and it was subsequently dropped from the EVC Computer Science curriculum.

In response to growing demand, EVC began to offer up to five (5) sections of Computer Science courses in the mid-90s. Thus, EVC decided to hire Harry Lichtbach as a full-time instructor. Lichtbach retired from EVC at the end of the 2010-2011 academic year. His tenure coincided with a period that has been labeled the *dot bomb*. It was during this period that Silicon Valley experienced a major downturn in the hiring of IT professionals and software engineers. According to data provided by our Institutional Effectiveness Committee (IEC), Evergreen Valley College did not offer more than two sections of Computer Science in any semester from the fall 2010 through the fall of 2013. ComSc 072 – Discrete Mathematics and ComSc 076 (Computer Science II) were offered every fall, and ComSc 075 (Computer Science I) every spring.

However, by the spring of 2014, growing student demand compelled EVC to offer four (4) sections in Computer Science. They included one section of Discrete Mathematics, two sections of Computer Science I, and one section of Computer Science II, for a total of 1.317 FTE. Because all the instructors for Computer Science courses were drawn from full-time and adjunct instructors in the Mathematics Department, the Vice-President of Academic Affairs decided to move the Computer Science program to the Division of Math, Science, and Engineering. The department then subsequently added a new course in Computer Organization and Systems (ComSC 77), offered for the first time in the spring of 2015. Altogether, the department offered six (6) sections of four courses in Computer Science in the spring of 2015 for a total of 2.017 FTE. But with no full-time instructor in the Computer Science department, qualified faculty from the Division of Math, Science and Engineering were called upon to teach these courses.

EVC has now added a fifth transferable non-majors course in Computer Science: Introduction to Computer Programming (ComSc 20). This course was offered for the first time this past summer (2015), and again this fall (2015) and has filled each time. For the spring of 2016, the department has scheduled two sections of ComSc 20, one section of ComSc 72, two sections of ComSc 75, one section of ComSc 76, and one section of ComSC 77, for a total of seven (7) sections, or 2.233 FTE. Note that all of these courses are transferable to the UC and CSU systems. As a result of this very rapid growth, the President of EVC has decided to hire a new full-time instructor in Computer Science. This instructor is expected to come on board in the fall 2016.

Courses in Computer Science prepare students for transfer to four-year colleges and universities, and the skills needed for some entry-level jobs in computer programming, as well as professional enrichment for incumbent workers and displaced workers. In addition, the discipline contributes to the Evergreen Valley College mission by providing:

- Courses that meet the requirements for transfer in Computer Science,
- Coursework that foster student growth and personal achievement, and
- An atmosphere that celebrates cultural diversity.

Note: For the following items, you may use Student Learning Outcomes (SLOs) and SLO assessments to gauge effectiveness. If so, you may wish to complete Part C, below, and then return to this section.

2. How does your program define effectiveness, and what measures have you chosen to gauge it?

- The program is effective to extent that students are successfully completing the required courses they need to transfer to a four-year college or university and pursue a baccalaureate degree in Computer Science or Software Engineering. This will be gauged by assessing rates of student retention, success, and persistence for students in each of our courses. The percentage of students who are transferring on to 4-year colleges and universities in computer science and software engineering will be obtained from the transfer center.

- The program is also thought to be effective if students who enroll in our courses feel welcome, independently of their race, culture, sexual orientation, or religion. This will be gauged for all courses in Computer Science by reviewing the students' responses to relevant questions on Student Evaluation Forms for Teaching Faculty, and the faculty peer Observation form for Classroom Faculty.

3. Please summarize the results of any measures you have applied. What do these results mean for your program?

Success rates for Computer Science students tend to be better than the college as a whole (see analysis A6). Information on the number of students transferring to 4-year colleges and universities in Computer Science is, for the moment, mainly anecdotal (from the students themselves). But the Computer Science program has been assured that the transfer center will be able to provide the needed data to more accurately assess students' transfer rates in the future. Note also that, more recently, quite a few of our students are taking just one or two of the four courses that make up the transfer program to enhance their work related skill sets.

4. Where would you like your program to be three years from now?

- Offer an AS-T in Computer Science
- Offer two additional elective courses that transfer to the UC and/or CSU, but are not among the required ones.

- Offer additional sections to evening students.
- Add hybrid and/or online courses to our offerings for students.
- Forge a partnership with our feeder high school districts to effectively recruit more students for our Computer Science program, and perhaps a summer "Boot Camp" in computer programming aimed at high schools students.
- Fill all the sections offered.

PART A: Overview of Program

1. Please state at least three recent accomplishments for your program which show how it contributes to the College's success.

The program added a new course to the curriculum in 2014 (ComSc 77 - Computer Organization and Systems) that has received C-ID approval. This capstone course is the last of four lower division courses that are required for all Computer Science majors. It was taught for the first time in the spring of 2015. Another new course, ComSc 20 (Introduction to Programming Concepts and Methodologies) was approved for C-ID in the spring of 2015. This is non-majors course with minimal mathematics prerequisites. It was offered for the first time in the summer of 2015.

One of the Instructors has been directing honors research projects for students in his discrete mathematics course. Some of the students presented their research in a conference sponsored by the *Mathematical Association of America* for college students in Northern California and Southern Oregon, and others presented their research to the students and faculty of Evergreen Valley College. These are part of the Department efforts in ensuring that our students are well prepared when they transfer to a university, enter the workforce, or embark on a path of lifelong learning, as stated in the District Mission Statement.

Another instructor wrote a proposal to the National Science Foundation to fund the development of a new Cybersecurity curriculum. Although the proposal was not funded, many elements of the proposal were subsequently included in the Silicon Valley Engineering Tech Pathways Consortium (SVETP) proposal, which was funded. EVC will receive $425,000 as a partner in this effort.

2. State the goals and focus of this department/program and explain how the program contributes to the mission, strategic initiatives, comprehensive academic offerings, and priorities of the College and District.

As noted in (1) above, the program has added 2 new C-ID approved courses, ComsSc 20 and ComSc 7, to the curriculum. ComSc 75 (Computer Science I) has been updated and received C-ID approval. The remaining two courses in the Computer Science curriculum, ComSc 72 (Discrete Mathematics) and ComSc 76 (Computer Science II), are currently undergoing revision and seeking C-ID approval as well. These changes address our District's strategic initiative to develop curriculum to support student achievement of their educational goals.

In addition, program faculty worked with the Enlace program to increase the number of Hispanic students who major in Computer Science and direct honors research projects, which have had a positive influence on our students' outlook on life-long learning. The goal of activities is to increase the number of students who feel valued and empowered.

Finally, one of our faculty played a leadership role in establishing partnerships with high schools, universities, governmental agencies, and for-profit and non-profit organizations, to develop a new curriculum in Cybersecurity. This project addressed our District's strategic initiative on community engagement.

3. Identify current student demographics. If there are recent changes in student demographics, explain how the program is addressing these changes.

From the fall of 2010 to the spring of 2013, the department typically offered two sections every fall and one section every spring. Beginning in the spring 2014, the department offered four sections of three courses for the first time in recent years. Moreover, our offerings increased to six sections of four courses in the spring of 2015. This growth has brought to light some interesting facts.

First the gender mix in Computer Science has always leaned heavily toward males. This has been of concern to Computer Science faculty for some time now, especially since females typically make up 53% to 55% of the general student population at EVC, and roughly 35% to 38% of the students enrolled in courses in Computer Information Technology (CIT). We should note that, unlike in CIT, students must be calculus-ready to enroll in Computer Science courses. Nevertheless, as is evident from the table below, the percentage of students that are female has grown with the number of sections offered by the department over the last two years. The program will explore a student club and other interventions to increase the number of women pursuing a major in Computer Science.

Gender	2013FA	2014SP	2014FA	2015SP
	Pct of Total	Pct of Total	Pct of Total	Pct of Total
Female	13.24%	16.67%	17.92%	19.84%
Male	85.29%	80.56%	82.08%	79.37%
No Value Entered	1.47%	2.78%		0.79%
Total Headcount:	**68**	**108**	**106**	**126**

As evident from the following table, the average age of students in the program is heavily skewed toward younger students, with a higher percentage of high school age students enrolling in the spring term. This undoubtedly reflects the fact that most our students are completing the first two years of a baccalaureate degree program. The increase in older (working age) students can be largely attributed to the two evening courses offered in the spring 2015.

Age	2013FA	2014SP	2014FA	2015SP
	Pct of Total	Pct of Total	Pct of Total	Pct of Total
17 & Below	1.47%	4.63%	0.94%	3.97%
18-24	82.35%	72.22%	79.25%	72.22%
25-39	11.76%	19.44%	17.92%	19.84%
40 & Over	4.41%	3.70%	1.89%	3.97%
Total Headcount:	**68**	**108**	**106**	**126**

Enrollment percentages among ethnic groups have remained fairly consistent over the last few years.

IPEDS Race Ethnic Classification	2013FA	2014SP	2014FA	2015SP
	Pct of Total	Pct of Total	Pct of Total	Pct of Total
Asian	66.18%	56.48%	64.15%	63.49%
Black or African American	1.47%	1.85%	2.83%	3.17%
Hawaiian/Pacific Islander				0.79%
Hispanic	13.24%	16.67%	14.15%	14.29%
Two or More Races	4.41%	4.63%	2.83%	3.97%
Unknown	7.35%	9.26%	4.72%	5.56%
White	7.35%	11.11%	11.32%	8.73%
Total Headcount:	**68**	**108**	**106**	**126**

4. Identify enrollment patterns of the department/program in the last 6 years and provide an analysis of any notable trends or patterns.

The two most notable changes in our enrollment over the last few years include an increase in the number of women (23.24% to 19.84%) and the number of students 25 and older (16.17 to 23.81%). This has occurred while the number of offerings in Computer Science has grown from two to four, then to six sections each semester. Moreover, in the spring of 2015, Computer Science offered two evening sections for the first time in quite a few years. These two courses in Discrete Mathematics and Computer Science 1 are foundational for the Computer Science major, and are, in part, an explanation for our changing demographics. It would appear that there is a large untapped market for evening students that we have just begun to reach.

5. (Identify department/program productivity (WSCH/FTEF).

	2013FA		2014SP		2014FA		2015SP	
	Measure	Pct Change from Previous Yr	Measure	Pct Change from Previous Yr	Measure	Pct Change from Previous Yr	Measure	Pct Change from Previous Yr
WSCH	280.8	NA	416	NA	421.6	50.14%	488	17.31%
FTES	8.557654	NA	12.677999	NA	12.848666	50.14%	14.872268	17.31%
FTEF	0.617	NA	1.317	NA	1.317	113.45%	2.0167	53.13%
Productivity	455.1053485	NA	315.8694002	NA	320.1214882	-29.66%	241.9794714	-23.39%

The department experienced a large drop in productivity over two years for which we have data. The productivity measure for the fall of 2013 measured 455.11, as compared to 324.39 for CIT classes, and 599.01 for the college as a whole. By the spring of 2015, the programs' productivity had dropped to 241.98 while the overall productivity of the college dropped by 11.5% to 482.11. Part of the explanation is that productivity measures for most of the college's programs appear to be seasonal, with the numbers up in the fall and back down in the spring. In the case of the Computer Science program, the other part of the explanation is that we offered ComSc 77 (Computer Organization and Systems) for the first time, and had a smaller enrollment than we expected. We also offered three sections of ComSc 75 (Introduction to Computer Science I). The additional section was taught at night for the first time in many years, and we had only than 16 students enroll in the course. We decided to let both courses continue in order to help build the Computer Science program. Indeed, by the fall of 2015, the evening offering of Computer I was waitlisted.

We have made what we believe to be the needed corrections. Instead of three sections of Computer Science I (two in the day and one at night), we will be offering one in the day and one at night (we did this in the fall of 2015, and both sections filled). Computer Organization and Systems is the capstone course for a Computer Science major, but until we can consistently fill the course, we will offer it only once a year in the spring semester at night.

6. Identify student success rate and patterns within the department/program paying particular attention to our college's target groups.

It is here that the sparseness of the data available to us becomes problematic. For example, we have no data on the percentage change from the previous years for success and completion rates in the fall of 2013 and the spring of 2014. What data we do have suggest that all groups are doing quite well relative the comparable college measures. Except for White students, all groups had very good success rates. In particular, success rates for Hispanics in the program ranged from 67% in the fall of 2013 to 89% in the spring 2015. These numbers compare favorably to the overall college success rates for Hispanics of 67% and 69%, respectively. It is important to note that most Computer Science major have already completed several courses in the calculus sequence, and are thus, proven problem solvers.

Success	2013FA	2014SP	2014FA		2015SP	
	Success Rate	Success Rate	Success Rate	Pct Change from Previous Yr	Success Rate	Pct Change from Previous Yr
American Indian						
Asian	74.47%	69.12%	71.01%	-4.64%	81.25%	17.55%
Black or African American	100.00%	100.00%	75.00%	-25.00%	100.00%	0.00%
Hawaiian/Pacific Islander					100.00%	
Hispanic	66.67%	83.33%	70.59%	5.88%	88.89%	6.67%
Two or More Races	100.00%	83.33%	100.00%	0.00%	57.14%	-31.43%
Unknown	50.00%	100.00%	100.00%	100.00%	100.00%	0.00%
White	100.00%	75.00%	86.67%	-13.33%	66.67%	-11.11%

Completion	2013FA	2014SP	2014FA		2015SP	
	Completion Rate	Completion Rate	Completion Rate	Pct Change from Previous Yr	Completion Rate	Pct Change from Previous Yr
American Indian						
Asian	85.11%	77.94%	81.16%	-4.64%	87.50%	12.26%
Black or African American	100.00%	100.00%	75.00%	-25.00%	100.00%	0.00%
Hawaiian/Pacific Islander					100.00%	
Hispanic	77.78%	83.33%	94.12%	21.01%	94.44%	13.33%
Two or More Races	100.00%	83.33%	100.00%	0.00%	57.14%	-31.43%
Unknown	50.00%	100.00%	100.00%	100.00%	100.00%	0.00%
White	100.00%	91.67%	86.67%	-13.33%	73.33%	-20.00%

7. If the program utilizes advisory boards and/or professional organizations, describe their roles.

Computer Science is a transfer program, thus this section is not applicable.

PART B: Curriculum

1. Identify all courses offered in the program and describe how the courses offered in the program meet the needs of the students and the relevant discipline(s).

Non-majors Course
Computer Science 020 – Introduction to Programming Concepts and Methodologies

Computer Science Majors Courses
Computer Science 072 – Discrete Mathematics
Computer Science 075 – Computer Science I: Introduction to Program Structures
Computer Science 076 – Computer Science II: Introduction to Data Structures
Computer Science 077 – Computer Organization and Systems

ComSc 020 is intended for students majoring in Business, Social Science, and the Biological Sciences, as well as students who have an interest in learning how to program a computer. It has no substantial mathematical prerequisite. It is also provides potential Computer Science students with a more gentle introduction to computer programming. It is currently taught with the Python programming language.

ComSc 072 provides the mathematical foundation for Computer Science (much like calculus provides the foundation for the physical sciences). Many of the California State Universities (CSUs) and the Universities of California (UCs), and in particular, San Jose State University and UC Berkeley, now required this course as preparation for students majoring in Mathematics as well as Computer Science.

ComSc 075 is the other foundational course for a computer science major. Some universities also require it for students majoring in the Physical Sciences, various Technology programs, Engineering, and Mathematics (STEM majors). This course includes basic computer science theory and Java programming.

Together with ComSc 072 a preparation for students transferring to the CSU or UC, and majoring in Computer Science, Engineering and Computer Science, or Software Engineering.

ComSc 072, ComSc 075, and ComSc 076 have been offered nearly every year that is included in the scope of this review. ComSc 077 was offered for the first time in the spring of 2015, and ComSc 020 was offered for the first time in the summer of 2015. Moving forward, our plan is to offer, every fall, two sections of ComSc 020 (day and night), one section of ComSc 072 (day), three sections of ComSc 075 (morning, afternoon, and night), one section of ComSc 076 (day), and one section of ComSc 077 (night). The spring will feature the following lineup: two sections of ComCs 020 (day and night), one section of ComSc 72 (night), three sections of ComSc 075 (morning, afternoon, and night), two sections of ComSc 076 (day and night), one section of ComSc 077 (day).

2. State how the program has remained current in the discipline(s).

There are no full-time faculty in the Computer Science department. But faculty that teach the classes have kept abreast of the relevant literature in Discrete Mathematics and Computer Science and/or have memberships in the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). Indeed, these two organizations have jointly promulgated the 2013 recommendation for the Computer Science curriculum in our nation's colleges and universities. These recommendations have greatly influenced the content of all the courses offered by the department. In addition, faculty are working to develop an AS-T in Computer Science. ComSc 020, ComSc 075, and ComSc

077 have already achieved C-ID approval. We are now working on getting approval for ComSc 076, and a new ComSc 080 – Discrete Structures for Computer Science, a course that will replace ComSc 072 in the major.

3. All course outlines in this program should be reviewed and revised every six years. If this has not occurred, please list the courses and present a plan for completing the process, including timelines and dates for each course.

   Every one of the five courses offered by the department has been updated within the last six years.

4. Identify and describe innovative strategies or pedagogy your department/program developed/offered to maximize student learning and success. How did they impact student learning and success?

We are fortunate that our students are largely computer literate when they begin our classes, and consequently, it is easy to incorporate technological innovation into our program. We have made use of Google groups and Moodle for posting all our classroom presentations and sample programs, as well as for forums support, interventions, and assignment screening. Web-based course management systems such as MyProgrammingLab and Enhanced WebAssign have been incorporated into some of our offerings.

5. Discuss plans for future curricular development and/or program (degrees & certificates included) modification. Use a Curriculum Mapping form as needed.

The next order of business for the department is to achieve C-ID approval for ComSc 076, and our new ComSc 080 course so we can develop a new AS-T in Computer Science. The major obstacles are unit totals for our Computer Science and our Physics courses. Nevertheless, students who successfully complete the core courses in Computer Science, Mathematics, and Physics for the major should have no trouble transferring to a four-year college or university.

Students should meet with their counselors as early as possible to develop an effective educational plan. The following is a recommended sample plan for the first two years of a Computer Science major:

SAMPLE TWO-YEAR PROGRAM FOR TRANSFER IN COMPUTER SCIENCE

First Year

Fall	Units	Spring	Units
ComSc 075	4	ComSc 076	4
Math 071	5	Math 072	5
GE*	6	GE	6
	-------		--------
Total	**15**	**Total**	**15**
Summer	**Units**		
GE**	0-6		

Second Year

Fall	Units	Spring	Units
ComSc 080	3	ComSc 077	3
Phys 004A	5	Phys 004B	5
GE*	6-9	GE	6-9
	---------		--------
Total	**14-17**	**Total**	**14-17**

A MORE GRADUAL APPROACH FOR STUDENTS NEW TO PROGRAMMING

First Year

Fall	Units	Spring	Units
ComSc 020	3	ComSc 075	4
Math 071	5	Math 072	5
GE	6-9	GE	6
	--------		--------
Total	**14-17**	**Total**	**15**
Summer	**Units**		
GE**	3-6		

Second Year

Fall	Units	Spring	Units
ComSc 076	4	ComSc 077	3
ComSc 080	3	Phys 004B	5
Phys 004A	5	GE	9
GE	3		
	--------		-------
Total	**15**	**Total**	**17**

(*) Either the General Education Breadth Pattern for the California State University (CSU GE) or the Intersegmental Education Transfer Curriculum (IGETC) for the University of California.

(**) Math 73 – Multivariable Calculus (5 Units), is a requirement for the Computer Science major at UC Berkeley. Students transferring to Berkeley and some of the other Universities of California might consider taking this course in the summer.

6. Describe how your program is articulated with the High School Districts, CCOC (if applicable), and/or other four year institutions. (Include articulation agreements, common course numbering etc.)

ComSc 072 (Discrete Mathematics), ComSc 075 (Computer Science I), ComSc 076 (Computer Science II) have long been articulated with San Jose State University and the University of California at Berkeley, as well as other CSUs and UCs. Recently developed courses like ComSc 020 (Introduction to Programming Concepts and Methodologies), and ComSc 077 (Computer Organization and Systems), which is now articulated with SJSU, have been approved for C-ID. We will soon be adding the course ComSc 080 (Discrete Structures for Computer Science) when it achieves C-ID approval.

7. If external accreditation or certification is required, please state the certifying agency and status of the program.

   This is not applicable because Computer Science is not a vocational program.

PART C: Student Learning Outcomes and Assessment

1. On the program level, defined as a course of study leading to a degree or certificate, list the Program Learning Outcomes (PLO), if applicable. Please note, you may be completing this program review as a department or discipline and do not offer any degrees or certificates. In this instance, please disregard this question.

   Computer Science does not offer a degree or certificates.

2. List or describe all assessment mechanisms you are using to evaluate course and/or program student learning outcomes. Please provide a link to all the course and/or program SLO assessment matrices.

   Assessment is done via quizzes, tests, programming assignments, a final programming project, and the final exam.

SLO Assessment for ComSc 20

SLO #1: Describe the software life-cycle, including design, development, documentation, styles, testing, and maintenance of computer programs.
Assessment Tool: Multiple-choice questions and a computer program on Test #1
Evaluation Timeline: Fall, 2015.
Assessment Results: Seventeen out of Twenty-four students (71%) had a passing score of 70% or more for this SLO.
Analysis/Action Plan and Timeline: Nineteen of these same twenty-four students were able to score 70% or more on a program that had to write from scratch during the time allotted for the exam. And thirteen of these same students scored 100% on their Test #1 programs. This was very good, given that the course assumes students are writing programs for the first time. Accordingly, we won't be changing anything here.

SLO #2: Explain what an algorithm is and the fundamental role it plays in programming.
Assessment Tool: Test
Evaluation Timeline: Spring, 2016

Assessment Results: The first test of the semester consists of multiple choice questions related to this objective and a program that implements a specific algorithm. Twelve of the fifteen students (80&) received a passing grade of 70% or higher.

Analysis/Action Plan and Timeline: These results are very good. No changes are required.

SLO #3: Write programs that employ the principles of structured programming and be able to design and test programs using currently accepted methodologies.

Assessment Tool: Ten Programming Assignments

Evaluation Timeline: The end of spring 2016

Assessment Results: Forthcoming

Analysis/Action Plan and Timeline:

SLO #4: Write programs using object-oriented design, and contrast the difference between object-oriented and procedural code.

Assessment Tool: Questions on the Final Exam, and a Final Programming Project

Evaluation Timeline: Fall 2015

Assessment Results: Seventeen of the twenty-two students (77%) received a passing grade of at least 70% the related questions on an objective exam, and 18 of the 22 students received a passing grade 70% on the final object-oriented programming project. Four students did not take the final exam and programming project.

Analysis/Action Plan and Timeline: The results are consistent with the good performance by the students throughout the term, and changes are being considered at this time. Our biggest concern is for the four students who did not take the final exam for this course.

SLO Assessment for ComSc 72

SLO #1: Analyze the logical structure of a proposition and that of a quantified predicate, and be able to conclude whether two quantified predicates are logically equivalent.

Assessment Tool: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall 2012.

Assessment Results: Six of the seven problems in the first test in the fall of 2012 were designed to assess various aspects of this SLO. Out of the 39 students who took the test, 32 (82%) received a passing grade of 60% or higher.

Analysis/Action Plan and Timeline: The results were satisfactory, and no change is being considered.

SLO #2: Recognize valid and invalid arguments, and be able to employ simple proof techniques such as proof by cases and proof by contradiction and contraposition.

Assessment Tool: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall 2012

Assessment Results: Three problems on the second test were designed to assess this SLO. Out of 32 students who took the test, only 16 (50%) received a passing grade.

Analysis/Action Plan and Timeline: Abstract proofs have always been difficult for students. We will look carefully at the included on this exam, and alert students to the need for more focused attention to these types of problems. We will also identify additional resources on-line and at our own Math and Science Resource Center to better assist student in the future.

SLO #3: Use truth tables and Boolean algebra to analyze logical circuits.

Assessment Tool: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall 2012

Assessment Results: Two problems on the second test were designed to assess this SLO. Out of 32 students, 26 (81%) received a passing grade. In fact, 21 of the students received perfect scores on both problems.
Analysis/Action Plan and Timeline: These results were very good, thus, no change is being considered.

SLO #4: Use simple modular arithmetic to solve problems, and be able to compute and convert numbers in different bases.
Assessment Tool: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall, 2012

Assessment Results: Six problems in the third test that fall were designed to assess this SLO. Out of the 27 students who took the test, 17 (63%) received a passing grade.

Analysis/Action Plan and Timeline: The results are about normal for students at this level. No changes are being considered at this time.

SLO #5: Use the principle of mathematical induction in proofs, and use recursion techniques in definitions and problem solving.

Assessment: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall, 2012

Assessment Results: Three problems were on the third test were designed to assess this SLO. Out of the 27 students who took the test, 11 (41%) received a passing grade.

Analysis/Action Plan and Timeline: this is another difficult topic for students at this level. The instructor is re-evaluating his presentation of the material and its pedagogy.

SLO #6: Solve simple counting problems and problems involving permutations and combinations.

Assessment: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall, 2014

Assessment Results: A problem consisting of two different parts on the final examination was designed to assess this SLO. Out of 24 students who took the final, 18 (75%) received a passing grade.

Analysis/Action Plan and Timeline: The results are about normal for students at this level. No changes are being considered at this time.

SLO #7: Use the concepts of sets, functions, relations, and equivalence relations in analyzing and solving mathematical problems.

Assessment: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall, 2013

Assessment Results: A problem consisting of two different parts on the final examination was designed to assess this SLO. Out of 31 students who took the final, 17 (55%) received a passing grade.

Analysis/Action Plan and Timeline: This is another difficult topic for students at this level. The course is now being re-structured and we will consider devoting more time to this topic when we offer this course in its new form.

SLO #8: Establish basic properties of graphs and use graphs as models in applications.

Assessment: Problems embedded in tests and the final examination.

Evaluation Timeline: Fall, 2014

Assessment Results:
Two problems each consisting of two different parts on the final examination were designed to assess this SLO. Out of 22 students who took the final, 19 (86%) received a passing grade.

Analysis/Action Plan and Timeline: These results are satisfactory for this SLO, and no change is being considered.

SLO Assessment for ComSc 75

SLO #1: Analyze problem descriptions, apply problem-solving methods, and use design tools to develop algorithms to solve simple problems.

Assessment Tool: Objective test and basic computer programming assignments.

Evaluation Timeline: Fall, 2015.

Assessment Results: Fifteen multiple-choice questions and a computer program were designed for the first exam of the term to assess this SLO. Forty-eight out of the sixty-six students (72 percent) received a passing grade of 70 percent or higher.

Analysis/Action Plan and Timeline: The results were satisfactory and no changes are being considered for the assessment tools.

SLO #2: Design code, debug, test and document programs that use the fundamental constructs: basic computation and sequencing, decision and iterative structures, and the definition of functions (methods).

Assessment Tool: Four elementary programming assignments.

Evaluation Timeline: Fall, 2015

Assessment Results: The assignments were designed to address sequencing, decision, and looping structures, together with a simple program that employed a method (function). 54 of the 66 students (81 percent) assessed had a passing score of 70% or better.

Analysis/Action Plan and Timeline: These results seem satisfactory, although, interestingly, I observed that slightly less than 80% of the students who begin this class, successfully complete it. Since this particular objective addresses the most basic part of computer programming, it behooves us put more focus on improving its measurement in future offerings of this course.

SLO #3: Identify and describe the properties of a variable such as its associated address, type, value, scope, persistence and size.

Assessment Tool: Multiple-choice questions and a computer program on the second exam.

Evaluation Timeline: Spring 2015

Assessment Results: Forty-five students took this exam, and thirty-three (73%) of the students scored 70% or better on this exam.

Analysis/Action Plan and Timeline: By and large, the students did well with the program, but a larger percentage had difficulty with the multiple-choice portion of the exam. A more thorough analysis revealed there were some questions on the exam that were missed by a majority of students because they did not understand what was being asked, i.e., the question was poorly written or the student was ESL. We have made what we believe are the necessary revisions for future measurement of this objective.

SLO #4: Implement arrays, strings, classes and objects in computer applications.

Assessment Tool: Five computer programming assignments covering arrays, strings, and object-oriented applications.

Evaluation Timeline: Spring 2015

Assessment Results: These were advanced programs worth a total of 100 points. They addressed, roughly, the last third of the course. Thirty-eight of the forty-five remaining students (82%) achieved a passing score of 75 or higher.

Analysis/Action Plan and Timeline: The passing threshold is little higher because the programs were not on a test. These results are in line with past offerings of this course. We must remember that most of these students have already completed courses in calculus, and managed to survive the first two thirds of the class. Hence, no changes are being considered here.

SLO #5: Develop programs to create and process sequential data files.

Assessment Tool: Two questions and a programming problem given on the first exam of the term.

Evaluation Timeline: Fall 2015

Assessment Results: Sixty-six students took the exam. Fifty-four (81%) of the students were able to score 70% or higher on this part of the exam. However, only 37 out of 66 students (56%) of the students answered the questions about sequential file processing correctly.

Analysis/Action Plan and Timeline: The latter score was a bit of a surprise, and not in line with the much better past results. We will be monitoring this topic more closely the next time we assess this SLO in the Fall 2016.

SLO #6: Design and implement classes to write object-oriented programs.

Assessment Tool: A capstone final programming project.

Evaluation Timeline: Fall 2014

Assessment Results: Thirty-nine of the forty-seven students (83%) achieved a passing grade of 70% or higher.

Analysis/Action Plan and Timeline: Actually most of the students did much better than 70% (closer to 100%). The problem had students design three classes, a base class named Employee, a subclass of Employee name ProductionWorker, and a subclass of ProductionWorker name TeamLeader. The assignment addressed some of the elements of object-oriented programming such as abstraction, encapsulation and inheritance. The students' very good performance on the project suggests that we should devise a final project that goes a little further, and include polymorphism, modularity, interfaces, and code reusability. We will be revising the project for the very next offering of this course, and will reassess it in the spring of 2016.

SLO #7: Summarize the history and evolution of programming languages, including paradigms in current use.

Assessment Tool: Multiple-choice exam

Evaluation Timeline: Fall 2015.

Assessment Results: Fifty-one of the sixty-six students (77%) who took the exam scored 70% or better on the part of the exam that covered this objective.

Analysis/Action Plan and Timeline: These results are in line with past exams and are acceptable, therefore, no actions are being considered at this time.

SLO Assessment for ComSc 76

SLO #1: Design and implement an Abstract Data Type

Assessment Tool: Programming Lab Assignment

Evaluation Timeline: Fall, 2015.

Assessment Results: Students were asked to design and implement a priority queue using a heap data structure in one of their labs. Almost 82% were able to achieve all the requested goals within the specified time frame. 14% of the students were one session late. Only one student who later dropped the class did not finish.

Analysis/Action Plan and Timeline: The results were satisfactory and no changes are being considered for the assessment at this time.

SLO #2: Compare and explain the efficiency of algorithms.

Assessment Tool: Midterm exam.

Evaluation Timeline: Fall, 2015

Assessment Results: Students were given a variety of questions to evaluate the running times of different algorithms. Of the 26 students that took the exam all but two demonstrated an understanding of the concept of efficiency of an algorithm.

Analysis/Action Plan and Timeline: This is an import concept in Computer Science that every student needs to understand. We suggest an extra assignment to help the students gain mastery of the topic.

SLO #3: Employ a disciplined approach to the design, coding, and testing of computer programs.

Assessment Tool: Six programming assignments.

Evaluation Timeline: Spring 2015

Assessment Results: The class started with 30 students. Two of the students eventually dropped the class. 25 (89%) of the remaining students were able to achieve an acceptable cumulative score of 80 or higher out of 100 points possible.

Analysis/Action Plan and Timeline: The assigned programs covered areas of object-oriented programming such as inheritance and polymorphism, exception handling, recursion, sorting and searching, generics, and collections. Many of the students did a very good job this semester on some conceptually difficult programming assignments. Nevertheless, there were some students who would benefit from more time working on this material directly with the instructor.

SLO #4: Design and write an application using both system defined and user defined ADTs; for example linked lists, stacks, queues and trees.

Assessment Tool: Programming Assignments

Evaluation Timeline: Spring 2016

Assessment Results: Eighty-two percent or 23 of the 28 students enrolled in the class achieved a score of 75% or better for four programs worth a total of 50 points.

Analysis/Action Plan and Timeline: The programs included linked lists, stacks, queues, and binary trees. Since students get to work on these programs at home as well as in our lab, a passing score is slightly higher than the usual 70%. Three of the five students who did not receive a passing grade on this objective were very close to a passing score. These results are in line with our previous offerings of this course, especially since data structures are introduced in the course after we have covered some advanced programming methodologies. Thus, no changes are recommended at this time.

SLO #5: Identify the different implementations of data structures covered in this course.

Assessment Tool: The Final Exam

Evaluation Timeline: Spring 2016

Assessment Results:

Analysis/Action Plan and Timeline:

SLO #6: Use sophisticated searching and sorting algorithms and understand their analysis.

Assessment Tool: Midterm exam.

Evaluation Timeline: Fall 2015

Assessment Results: Students were tested on the advanced sorting algorithms such as Bucket Sort, Counting Sort, Radix Sort, and Quick Sort. 89% of the of the 26 students were able to perform their running time analyses successfully

Analysis/Action Plan and Timeline: These results appear to be satisfactory. We have no changes to recommend at this time.

SLO Assessment for ComSc 77

SLO #1: Describe the architectural components of a computer system.

Assessment Tool: Quizzes and computer programs.

Evaluation Timeline: Spring 2017.

Assessment Results:

Analysis/Action Plan and Timeline:

SLO #2: Discuss and demonstrate the use of compilers, linkers, and loaders.

Assessment Tool: Quizzes and computer programming assignments.

Evaluation Timeline: Spring, 2017

Assessment Results:

Analysis/Action Plan and Timeline:

SLO #3: Describe computer representation of numbers and how computer arithmetic is carried out.

Assessment Tool: Ten Programming Assignments

Evaluation Timeline: Spring 2016

Assessment Results: Students were given a programming assignment and a quiz on this objective, and 82% had an excellent performance while 18% did very well.

Analysis/Action Plan and Timeline: No changes recommended

SLO #4: Describe the representation of nonnumeric data (character codes, graphical data).

Assessment Questions on the Final Exam, and a Final Programming Project

Evaluation Timeline: Spring 2017

Assessment Results:

Analysis/Action Plan and Timeline:

SLO #5: Demonstrate knowledge of the MIPS assembly language.

Assessment Tool: Quizzes and computer programming assignments.

Evaluation Timeline: Spring 2016

Assessment Results: Sixty-seven percent of the students did excellent work, 26% did very well, and 7% did an acceptable job.

Analysis/Action Plan and Timeline: No changes are recommended for this SLO.

SLO #6: Compare and contrast MIPS architecture and assembly language with IA32.

Assessment Tool: Quizzes and programming assigjnments

Evaluation Timeline: Spring 2017

Assessment Results: Sixty-seven percent of the students did excellent work, 26% did very well, and 7% did an acceptable job.

Analysis/Action Plan and Timeline: No changes are recommended or this SLO.

SLO #7: Write and debug assembly programs that use load/store, arithmetic, logic, branches, call/return and push/pop instructions.

Assessment Tool: Quizzes and programming assignments

Evaluation Timeline: Spring 2016

Assessment Results: Sixty-seven percent of the students did excellent work, 26% did very well, and 7% did an acceptable job.

Analysis/Action Plan and Timeline: No changes are recommended for this SLO.

SLO #8: Discuss how variable access, arithmetic, function calls, and pointers are translated from a high level language into assembly.

Assessment Tool: Quizzes and programming assignments

Evaluation Timeline: Spring 2016

Assessment Results: Sixty-seven percent of the students did excellent work, 26% did very well, and 7% did an acceptable job.

Analysis/Action Plan and Timeline: No changes are recommended for this SLO.

SLO #9: Write programs that interface between a high level language and assembly.

Assessment Tool: Quizzes and programming assignments.

Evaluation Timeline: Spring 2017

Assessment Results:

Analysis/Action Plan and Timeline:

SLO #10: Write programs that contain system calls.

Assessment Tool: Quizzes and programming assignments.

Evaluation Timeline: Spring 2017

Assessment Results:

Analysis/Action Plan and Timeline:

SLO #11: Demonstrate and evaluate the use of efficient programming techniques.

Assessment Tool: Quizzes and programming assignments.

Evaluation Timeline: Spring 2017

Assessment Results:

Analysis/Action Plan and Timeline:

3. Since your last program review, summarize SLO assessment results at the course and program level (if applicable).

   Not applicable. There were no assessments of Computer Science courses completed in prior program reviews.

4. What plans for improvement have been implemented to your courses or program as a result of SLO assessment?

The content and SLOs for ComSc 75 – Computer Science I: Program Structures were revised in 2013, and the course received C-ID approval. The department began to offer the new version of this course beginning in the fall of 2014. Its assessment is now complete, and the results were very good. No changes have been planned for this course at this time.

ComSc 77 – Computer Organization and Systems, received C-ID approval in 2014, and we offered the course for the first time in the spring of 2015. The adjunct faculty member who taught the course that spring did not provide the department with an assessment of any of the objectives. The course will be offered once a year, every spring semester. Five objectives were assessed in the spring of 2016, and the remaining six will be assessed in the spring of 2017.

ComSc 20 – Introduction to Programming Concepts and Methodologies, received C-ID approval early in 2015. It was offered for the first time in the fall of 2015. We are planning to offer this course every semester.

ComSc 76 – Computer Science II: Data Structures is in the process of being revised. The content and SLOs for this course are being changed to achieve C-ID approval. The assessment of the existing course will be complete when one final SLO is assessed in the spring of 2016. This last SLO is will effectively be part of the new version of this course.

ComSc 72 – Discrete Mathematics, has been completely assessed and the department is agreed that it is in need of changes. A large part of the problem is that the course's sole prerequisite is Precalculus Algebra and Trigonometry. There is no programming prerequisite. The department has made the decision to develop a new course, ComSc 80 – Discrete Structures for Computer Science, which will have ComSc 75 as prerequisite. Thus, students who enroll in this new course will have more mathematical maturity, and greater problem solving skill when they encounter this material. These changes will also allow us to apply for C-ID approval. Moreover, once this course is in place, the department be in a position to develop a new AS-T in Computer Science.

5. As a result of SLO assessment data, will you be requesting additional resources for your program or courses (i.e. additional faculty, equipment request, program personnel…)?

Henry Estrada will have a 100% instructional assignment in the Computer Science Department effective fall 2016. However, Chungwu Ho is retiring at the end of the spring 2016, and we will have to identify qualified adjunct faculty to teach additional sections. The program will also require instructional assistance for On-line and Hybrid courses (support, screening, and tutors). Finally the Computer Science Department will need to purchase annual software licenses for TextPad (for Java Programming) and MATLAB.

PART D: Faculty and Staff

1. List current faculty and staff members in the program, areas of expertise, and how positions contribute to the program success.

Estrada, J. Henry

<u>Area of Expertise:</u>

Professor Estrada's formal education was in Math, Physics, and Computer Science. And although he has taught every college level Mathematics and Computer Science and course offered at EVC, his recent focus has been on teaching Calculus, Linear Algebra, and Computer Science I & II.

<u>How Does His Position Contribute to the Program Success?</u>

Professor Estrada was the principal architect for our A.S. Degree for Transfer in Mathematics, which was approved in 2012. The new degree provides the foundational training that students need to meet their higher educational goals in Mathematics and Applied Mathematics. Since 2012, he has been actively engaged in getting high-tech companies, government agencies, universities and public schools to partner with EVC to develop curriculum in Cyber Security. He was the principal investigator on grant proposals to the National Science Foundation, and other funding agencies concerned with Cyber Security. In the past two years, he has updated the curriculum for Computer Science 75 (Introduction to Program Structures) and is now in the process of updating Computer Science 76 (Introduction to Data Structures). The principal changes to these courses included changes to content and SLOs. The changes made to Computer Science 75 were essential as the course in an elective in our A.S. Degree for Transfer in Mathematics. Finally, he has added two new courses to our computer science curriculum: Computer Science 20 (Introduction to Programming Concepts and Methodologies) and Computer Science 77 (Computer Organization and Systems). Computer Science 77, which was offered for the first time in the spring of 2015, serves as a capstone course for students planning to transfer to universities and major in computer science, software engineering, and computer engineering. The Computer Science 20 course was designed as a service course for non-majors, but also as a way of attracting more students into Computer Science. Professor Estrada will be teaching this new course in the fall of 2015. He will be responsible for assessing ComSc 20, ComSc 75, and ComSc 76.

Ho, Chungwu

Area of Expertise:

Professor Ho focuses on upper level mathematics, discrete mathematics, applications of mathematics, and working with individual honor students on research projects.

How Does His Position Contribute to Program Success?

Professor Ho has been responsible for the course content and learning objectives for our Discrete Mathematics course. He has been the faculty member assessing the SLOs for Discrete Math. He has also coordinated the AS-T SLO assessments for the courses multivariable calculus and differential equations in the Mathematics department. These are part of the Department efforts to provide quality and efficient programs to ensure student success, as delineated in the School Strategic Initiatives, and to ensure that our students are well prepared when they transfer to a university or to enter the workforce, as stated in the District Mission Statement "*to meet the diverse educational and workforce needs of our community by empowering our students* ...".

He has also been directing students' honor research projects related to multivariable calculus, differential equations, and discrete math courses. For the past six years, 33 students completed their honor's research projects under his supervision, and six of them presented their research in a conference sponsored by the *Mathematical Association of America* for college students in Northern California and Southern Oregon, and 12 others presented their research to the students and faculty of Evergreen Valley College. These are part of our Division's efforts to ensure that our students are well prepared when they transfer to a university, enter the workforce, or embark on a path of lifelong learning, as stated in the District Mission Statement.

Professor Ho has been working extensively with students of diverse background: Among the 40+ students who have completed their honor's research projects under his supervision, there were, in addition to white, Hispanic, and Asian American students, also students from Vietnam, China, India, Pakistan, Cambodia, and Thailand. This is a part of the department efforts of empowering and preparing students from diverse backgrounds to succeed academically, as stated in the School Mission Statement.

Eftekhari, Kamran

Area of Expertise

Dr. Efteckhari is an Adjunct faculty member in the Division of Mathematics, Science, and Engineering. His formal education was in Mechanical Engineering. He has taught courses in Computer Science, Computer Information Systems, and CAD/CAM at several community colleges, including Evergreen Valley College.

How Does His Position Contribute to Program Success?

Although he has taught several of our courses in Computer Science, his principal contributions and responsibilities are to teach and assess SLOs for ComSc 76 – Computer Science II, and ComSc 77 – Computer Organization and Systems. He is also a member of the Institute for Electrical and Electronics Engineers (IEEE).

2. List major professional development activities completed by faculty and staff in this department/program in the last six years and state proposed development and reasoning by faculty in this program.

Estrada, J. Henry

Professional Development in the Last Six Years:

Since most of the students he encounter in his classes are in STEM fields, Professor Estrada makes an effort to stay current in applications of mathematics and computer science to several areas of science and technology. He has attended all of the training sessions at EVC on SLO development and assessment. He has also attended workshops and conferences on topics such as sustainable energy systems, smart grids, and cybersecurity (with emphasis on data encryption). More recently, he has attended several conferences and workshops sponsored by Homeland Security, SRI International, and the NSF funded CyberWatch West. He was the principal investigator on a grant proposal to the National Science Foundation. Although it was not funded, major elements of the proposal were included in the Silicaon Valley Engineering Tech Pathways (SVETP) proposal that was funded for $13 million. In the years to come, we can expect the cybersecurity space to employ an increasing number of applied mathematicians and computer scientists. He keeps current on developments in Computer Science through his membership in the Association for Computing Machinery (ACM).

Proposed Professional Development Activities and Reason for such Activities:

He expects to focus on further developing our curriculum in Computer Science, particularly, SLOs and their assessment. He will make every attempt at remaining current on methods of SLO implementation. Professor Estrada will also continue to work toward developing an A.S. Degree for Transfer in Computer Science, and once we have an established program in Computer Science, he plans to plans to develop hybrid and online versions of the courses that are offered by the department.

Ho, Chungwu

Professional Development in the Last Six Years:

He has been learning the C-ID course requirements for Comsc 72, and methods for assessing the SLOs for the course. Professor Ho participated in several workshops on MS Office and Moodle. He has also been engaged in research and published a paper in the journal, *Mathematics Magazine of the Mathematical Association of America,* and has published two poems in the literary magazine, *Leaf by leaf*.

To keep abreast with the advancement of knowledge, he has been a member of the *American Mathematical Society* and the *Mathematical Association of America*. Professor Ho is also a former associate member of the *American Academy of Poets*.

Proposed Professional Development Activities and Reason for such Activities:

Professor Ho will continue his efforts in updating and assessing the SLO of the courses he will be assigned to teach and expanding his knowledge in discrete mathematics so that he can bring in meaningful applications of mathematics to his classes and interesting topics for student research.

3. Identify current schedule for tenure review, regular faculty evaluation, adjunct faculty evaluation, and classified staff evaluation.

Evaluation and Tenure Schedule of Non-Tenured Faculty

Non-tenured faculty members in the past six years have been evaluated by following very closely the procedure described in *Article 20* of the *FACBA*. At the beginning of the non-tenured member's first semester a *Tenure Review Committee* (*TRC*) is formed according to the selection procedure delineated in section *20.2* of the *FACBA*: a committee consisting of the Dean, and two tenured faculty members, one of which is selected by the faculty being evaluated and the other selected by the Dean.

In the first three years of service for non-tenured faculty, a *Pre-evaluation Conference* is convened by the end of the ninth week of the non-tenured faculty member's first semester, and by the end of the fifth week of the non-tenured member's third and fifth semesters.

The faculty member's classes are then visited and student evaluations are collected by the *TRC* members. The non-tenured faculty member also makes a *Self-evaluation* and designs a *Growth and Development Plan* according to *FACBA 20.8.2*. A *Progress Review Conference* is convened by the end of the fourteenth week of the non-tenured faculty member's first, third and fifth semesters to review the information from the *TRC* members, student evaluations and the faculty's *Growth and Development Plan*. A *Post-Evaluation Conference* is convened by the fourth week of the non-tenured faculty member's second, fourth, and sixth semesters to review and finalize the non-tenured faculty member's *Growth and Development Plan.*

In the fourth year, the *Pre-Evaluation, Progress Review Evaluation*, and the *Post-Evaluation Conferences* are all completed by the end of the non-tenured faculty member's seventh semester. At the fourth year, the TRC chair drafts a *Summary Evaluation Report* based on the classroom observations, administrator and student evaluations, job description and the non-tenured faculty member's *Growth and Development Plan* and a tenure recommendation is submitted to the College President.

Evaluation of Tenured Faculty Members

The department evaluates tenured faculty members once in every three years according to the procedure set in *FACBA Article 22.* The dean first informs the faculty being evaluated in advance of the procedure and criteria of evaluation. The faculty being evaluated makes a *Self-Evaluation* according to *FACBA 22.2.4* and designs a *Growth and Development Plan* according to *FACBA 22.7.* An Evaluation Committee (*EC*) is then established, which consists of the dean and a tenured faculty member. A *Pre-evaluation conference* is then held for the faculty member with the EC members. The committee reviews and modifies the Growth and Development Plan, and schedules the classroom observations. After the EC members visit the classes and collect the student evaluations a *Post- Evaluation Conference* is then held, in which the EC members evaluate the faculty member's performance according the criteria delineated in *FACBA 22.6*, reviews and finalizes the faculty member's Growth and Development Plan, and complete the *Summary Evaluation Report*.

If the faculty member agrees with the report, it is then finalized and signed by the EC members and the faculty member. If the faculty being evaluated does not agrees with the summary, he/she will be given an opportunity to make a written response, which will be considered as a part of the final evaluation report.

Evaluation of Adjunct Faculty

Adjunct faculty have been evaluated according to the procedure as spelled out in Article 19 of *Faculty Association Collective Bargaining Agreement* (*FACBA*). An evaluation committee is formed, which usually consists of the Dean of the Division and one or two full-time faculty members. The adjunct faculty is fully informed in advance of the procedures of the evaluation process and the criteria upon which evaluation are conducted. At least one, sometimes all, of the committee members observe the performance of the adjunct faculty member, using the criteria for evaluation listed in the *FACBA 19.3.*During the observation, student evaluations are conducted and collected by the committee member. The adjunct faculty member is given a written summary of these evaluations and a post evaluation conference is held with the adjunct faculty and the evaluation committee. The evaluation for adjunct faculty has been carried out according to the status of the faculty: those who have achieved seniority rehire preference (SRP) are evaluated every sixth semester, those who are assigned 33% or more load are evaluated in each of three consecutive semesters, and those working less than 33% without SRP are evaluated in the first semester of employment and thereafter every sixth semester of employment.

4. Describe the departmental orientation process (or mentoring) for new full-time and adjunct faculty and staff (please include student workers such as tutors and aides).

The Departmental Orientation Process for New Faculty

The District Orientation for new full-time faculty is described in *FACBA 5.13.1*, which usually takes place on or near the day before the first Professional Development Day of the fall semester for a maximum of 6 1⁄2 hours, and the Faculty Union will have up to 3 hours of the orientation meeting. In addition to this District Orientation Program, the Division also has a mentor program: each of the new faculty members, in his/her first semester of service, is assigned a tenured faculty as his/her mentor, who provides guidance and assistance to the new faculty member to fit into the school environment, assisting the new member in questions related both to students (such as admission and registration procedures, adding and dropping classes, etc.) and faculty (such as tenure procedure, teaching assignments, and committee work).

The Departmental Orientation Process for Adjunct Faculty

The District also has an orientation program for new adjunct faculty is described in *FACBA 5.13.2*. New adjunct faculty will receive two hour pay for attending the school adjunct orientation, including a 30 minutes orientation with the Faculty Union. In addition, when an adjunct faculty is hired, he or she is provided with the learning objectives for the class the faculty member is assigned to teach and the syllabus used by our current faculty members. The dean or a current faculty member also describes in detail how our classes are conducted, together with student matters such as adding and dropping students, attendance policy, etc.

The department is challenged by the maintenance and improvement (professional development) of full-time faculty, as well as staff and administrative levels to support instructional needs and student support services and keep abreast with recent retirements. Indeed, to sustain current levels of service, the college must commit to a staffing plan, linked to resource allocation, which analyzes human resource needs based upon the size, scope, and changing needs (demographics shifts and gender gaps) of the department: it is then a good idea, for the mathematics department, guided by more extensive student data and by the college and district educational master plans, to assess and analyze the level and diversity of its full-time faculty and staff. The math department and the college could then use the results of that assessment to develop, adopt, fund, and implement long-range staffing that will ensure a sufficient number of qualified and diverse fulltime faculty, part time faculty and staff (including tutors) to foster an equitable and inclusive environment for all students and assure the quality of the program. Gender gaps in full time faculty and in STEM courses need to be addressed promptly.

PART E: Facilities, Equipment, Materials and Maintenance

1. Identify and discuss the facilities, equipment, equipment maintenance, and materials allocated to the program. Identify and explain additional facility needs and rationale.

As Computer Science was recently part of the Division of Business and Workforce Development all of our sections made use of the computer labs in rooms RF-234 and RF-241. RF-234 has Python and Java interpreters installed so that students can write their programs for ComSc 020 and ComSc 075. RF-214 has a Java interpreter (TextPad), as well as an assembler, Linux, and a C compiler installed for use by ComSc 075, ComSc 076, and ComsCs 077. Each of these labs has 30 computer workstations, which were upgraded during the 2014-2015 academic year.

The specifications of the computers are as follows:

- RF-241: Intel i7-3770 CPU, 3.4 GHz, 8 GB of RAM, 500 GB hard drive
- RF-234: Intel i5-3570 CPU, 3.4 GHz, 4 GB of RAM, 500 GB hard drive

Room RF-234 has a pay-for-print station; RF-241 has a standard printer for general student use.

The Computer Science Department offered six sections in the spring of 2015, and we were able to accommodate this growth of sections in these two labs alone. We are thankful that, for the sake of continuity, the Division of Business and Workforce Development was willing to accommodate our courses by sharing these labs with us even though Computer Science was now under the Division of Math, Science, & Engineering.

With the addition of the new ComSc 020 (taught in Python), the Computer Science Department planned to offer seven (7) sections in the fall of 2015. We quickly discovered that the Division of Business and Workforce Development had no room to accommodate an additional class and consequently the Computer Science Department had to work with Library staff to schedule our Tuesday, Thursday evening ComSc 075 in room LE-204. This was the only computer lab large enough to accommodate a class of 35 students.

The Computer Science Department is scheduled to move to new South Campus Building for the 2016-2017 academic year, and will have to share seven classrooms (60 seating capacity) and a computer lab (41 student seating capacity. The department will be sharing these resources with the Departments of Mathematics, Chemistry, and Physics. But the Department of Computer Science is planning to offer six sections in the fall of 2016 and seven sections in the spring of 2017, finding labs may once again prove problematic. Indeed, Computer Science was moved to the MS&E Division after the planning for our new building, and we suspect that we may need to add a new computer lab with seating for 40+ sometime in the next 4 years.

2. Describe the use and currency of technology used to enhance the department/program. Identify projected needs and rationale.

Desktop computer technology is adequate for the department's current needs, and also for the forseeable future. As mobile computing becomes more widespread, tablets may be an additional future option. Some of our courses could be scheduled as hybrid courses with lecture in the classroom and virtual labs on the Internet. Finally, we need to offer some of our courses as on the Internet.

3. If applicable, describe the support the program receives from industry. If the support is not adequate, what is necessary to improve that support?

   Not Applicable.

PART F: Future Needs

1. Current Budget
   A. Identify the budget currently allocated for the department/program through the division budget (fund 10). Discuss its adequacy in meeting your program's needs.

      The fund 10 budget for Computer Science for the 2015-16 academic year was $216. Recall that this is a program which has grown from two (2) sections offered in the Fall 2014 to seven offered in the spring of 2016. The program has requested $2,450.00 for 2016-2017:

Conferences	800.00
Printing and Reprographics	250.00
Supplies Instruction	400.00
Software over $200	1,000.00

B. Identify any external (fund 17) funding the department/program receives, and describe its primary use.

The Computer Science budget for this fiscal year is:

17-21 Equipment > $5000	5,795
Materials Fees	216
Total:	$6,011

C. Explain any grants or other external funding sources (partnerships) for which your program is benefiting from.

The program will apparently be receiving funds (to be determined) as a partner in the Silicon Valley Engineering Tech Pathways (SVETP) grant that was awarded to our District.

2. Explain any grants or other external funding sources for which your program would be a good candidate. Do you have plans to apply for such sources?

The best choice would be to apply for an NSF-ATE funded grant in Computer Science education. I plan to apply for a grant as soon as we have the Computer Science program on track.

3. Please describe any unmet needs for your program and how you plan to address them. Are any additional resources needed to accomplish your program's outcomes?

The Computer Science department needs support for online education, including technical support and professional development for students and faculty. The department will also need funding for annual licenses and maintenance for instructional software such as TextPad ($655.00 for 40 site licenses) and MATLAB ($750.00 for 10 site licenses).

4. What faculty positions will be needed in the next six years in order to maintain or build the department? Please explain. What staff positions will be needed in the next six years in order to maintain or build the department? Please explain.

Computer Science will have a new full-time instructor in the fall of 2016. Given the potential for rapid growth in this area, it is highly probable that the department will need a second full-time instructor within 4 years.

5. Does your program require any additional facilities, equipment, and/or supplies over the next six years (above and beyond the program's current budget)?

If the program begins to grow as rapidly as we are projecting, the college will need to acquire the additional computer hardware to support that growth. That would entail a new lab with 41 Dell All-in-one Desktop computers at a cost of approximately $70,000.00.

PART G: Additional Information

Please provide any other pertinent information about the program that these questions did not give you an opportunity to address.

The department is moving in the direction of offering an AS-T in Computer Science. The major stumbling block is the fact that our Calculus and Physics classes are currently all five semester units each (resulting in a unit total that exceeds the maximum 60 units for an AS-T) . We will be able to have an AS-T in Computer Science when Physics 004A and 004B are changed from 5 to 4 units, or the criteria for the TMC in Computer Science is changed. Computer Science will begin doing its part by reducing the units for ComSc 77 and ComSc 80 from 4 to 3 units.

PART H: Annual Assessment: Program Faculty and PR Committee

Please attach copies of any Annual Reviews that you have completed in the last six years (if applicable)

PART I: Resource Allocation Table

Program Reviews provide a valuable source of information for the College as it makes decisions on resource allocation, both in terms of funding and cuts. The following information, in table format, will be used by the College Budget Committee to help inform EVC's Budget and Planning Process.

Item Title	Response
Productivity (WSCH/FTEF)	281
Student Success Rate (Retention Rate)	78%
Number of class sections offered by your program	6
Changes in enrollment	17.31% over last academic year
Your Program's Current Budget (from Fund 10)	$216
Current External Funding (from Fund 17)	$6,011
Future Needs: Faculty (Estimated Additional Cost)	$130K
Future Needs: Staff (Estimated Additional Cost)	$40K (lab assistants & tutors)
Future Needs: Facilities (Estimated Additional Cost)	$71,405 (TextPad, MATLAB, and 41 Dell All-in-One desktops
Future Needs: Supplies (Estimated Additional Cost)	$6000.00

**** Do your program's future needs assume that your program's enrollment will remain stable or do they depend upon enrollment growth? If they depend on growth, please explain the growth projections on which you are basing your assumptions. You may attach any supporting documentation to explain or support assumptions.***