

Oregon Institute of Technology
Computer Systems Engineering Technology Department
Software Engineering Technology Program Assessment Plan
2010-2011

I. Introduction

The Software Engineering Technology (SET) program was implemented in 1984 and was initially accredited by TAC of ABET in 1991. The Portland program was established in Fall 1996 under the same accreditation. The Associate degree was accredited by TAC of ABET in 2009. The program has continuously evolved as industrial changes have warranted.

II. Mission, Objectives and Student Learning Outcomes

The program faculty reviewed and approved, with a few changes, the mission, objectives and student learning outcomes for the program during the first week of fall term. The mission statement, objectives and program outcomes for the baccalaureate program are located on the OIT website at www.oit.edu/provost/learningoutcomes/cset/swbs. The associate program's mission statement, objectives and program outcomes are located at www.oit.edu/provost/learningoutcomes/cset/swae.

In June of 2009 the Program Educational Objectives (PEOs) were rewritten at the request of ABET evaluators. These objectives for both the AE and the BS programs were approved by the departmental Industrial Advisory Board (IAB) during the meeting on November 18, 2009.

The new PEOs were still deemed by ABET evaluators to be inappropriate because the AE and BS PEOs were too similar. The AE PEOs were re-written to more accurately describe the associate degree objectives. These new PEOs were approved by the SET program faculty on September 22, 2010. The PEOs were then sent to the CSET Industrial Advisory Board for approval. The IAB members approved the new PEOs on October 8, 2010.

Bachelor Program Mission

The mission of the Software Engineering Technology (SET) Bachelor's Degree program within the Computer Systems Engineering Technology (CSET) Department at Oregon Institute of Technology is to prepare our students for productive careers in industry and government by providing an excellent education incorporating industry-relevant, applied laboratory based instruction in both the theory and application of software engineering. The program is to serve a constituency consisting of our alumni, our employers, and our Industrial Advisory Board. Major components of the SET program's mission in the CSET Department are:

- I. To educate a new generation of Software Engineering Technology students to meet current and future industrial challenges and emerging software trends.
- II. To promote a sense of scholarship, leadership, and professional service among our graduates.

- III. To enable our students to create, develop, apply, and disseminate knowledge within the software development environment.
- IV. To expose our students to cross-disciplinary educational programs.
- V. To provide government and high tech industry employers with graduates in software engineering and related professions.

Bachelor Program Educational Objectives

The Program Educational Objectives of OIT's Software Engineering Technology program are to produce graduates that:

- A. Use their knowledge of engineering to creatively and innovatively solve difficult computer systems problems.
- B. Regularly engage in exploring, learning and applying state-of-the-art hardware and software technologies to the solution of computer systems problems.
- C. Will be an effective software development team member that contributes innovative software design solutions to the resolution of business, scientific or government computer systems problems.
- D. Will communicate effectively and successfully, both individually and within multi-disciplinary teams.

Bachelor Program Student Learning Outcomes

Software Engineering Technology baccalaureate graduates will have demonstrated:

1. an ability to identify, formulate, and solve software engineering problems, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements (Program Objective A, B, and C);
2. the ability to elicit, analyze and specify software requirements through a productive working relationship with various stakeholders of the project (Program Objective A and C);
3. an understanding of the core areas of software engineering. (data structures, theory of computation, operating systems, compilers, programming languages, computer architecture). (Program Objective A);
4. an ability to function effectively on teams (Program Objective A and C);

5. an understanding of professional, ethical and social responsibility (Program Objective C);
6. a recognition of the need for, and an ability to engage in life-long learning (Program Objective C);
7. knowledge of and ability to apply discrete math, probability and statistics (Program Objective B);
8. the successful design, development, deployment and maintenance of a major software based project (Program Objective A, B and C);
9. an ability to convey technical material through oral presentation and interaction with an audience (Program Objective A and C);
10. an ability to convey technical material through written reports which satisfy accepted standards for writing style (Program Objective A and C);
11. an ability to evaluate the impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues and emerging software engineering trends, models, tools, and techniques (Program Objective A and C);

Associate Program Mission

The mission of the Software Engineering Technology (SET) Associate Degree program within the Computer Systems Engineering Technology (CSET) Department at Oregon Institute of Technology is to prepare our students for entry level careers in the software industry and government by providing applied laboratory based instruction. The program is to serve a constituency consisting of our alumni, our employers, and our Industrial Advisory Board. Major components of the SET program's mission in the CSET Department are:

- I. To provide a new generation of Software Engineering Technology students with a solid background in computer programming.
- II. To enable our students to create, develop and apply knowledge within a technical software environment.
- III. To provide government and high tech industry employers with entry level graduates in computer programming and related professions.

Associate Program Educational Objectives

The Program Educational Objectives of OIT's Software Engineering Technology program are to produce graduates that:

- A. Assist in solving computer systems problems using their knowledge of computer programming.

B. Regularly engage in learning and applying state-of-the-art hardware and software technologies to the solution of computer systems problems

C. Will communicate effectively and successfully in the workplace.

Associate Program Outcomes

Software Engineering Technology associates graduates will have demonstrated:

1. an ability to identify, formulate, and solve computer programming problems, including the specification, design, implementation, and testing of programs that meet specification, performance, maintenance and quality requirements (Program Objective A, B, and C);
2. an understanding of the core areas of software engineering (data structures and programming languages). (Program Objective A);
3. an understanding of professional, ethical and social responsibility (Program Objective B);
4. a recognition of the need for, and an ability to engage in life-long learning (Program Objective B);
5. the successful design and development of a computer program (Program Objective A, B and C);
6. an ability to communicate through oral presentation and interaction with an audience (Program Objective B);
7. an ability to convey technical material through written reports which satisfy accepted standards for writing style (Program Objective B);

III. Three-Year Cycle for Assessment of Student Learning Outcomes

The department assesses the program educational objectives and student learning outcomes on a three-year cycle. During the six-year ABET cycle, the objectives and learning outcomes will thus be fully assessed twice. The program faculty met during the first week of fall term and rearranged the assessment cycle to be more aligned with the Institutional Student Learning Outcomes (ISLOs). Specifically the outcome dealing with the global society (BS outcome #11) was moved to 2010-2011. To accommodate this change outcome #1 for the bachelor degree and the associate degree was moved to 2009-2010.

All appropriate accreditation documents are housed on a SharePoint site maintained by the department. All department members have access to this site, but the documents are not viewable by the general public. The public can view the baccalaureate outcomes at www.oit.edu/provost/learningoutcomes/cset/swbs and the associate outcomes at www.oit.edu/provost/learningoutcomes/cset/swbs.

Bachelor Degree Assessment Cycle

Table 3-1: Baccalaureate Outcome Assessment Timeline

#	Learning Outcomes	10-11	11-12	12-13
1	an ability to identify, formulate, and solve software engineering problems, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements			X
2	the ability to elicit, analyze and specify software requirements through a productive working relationship with various stakeholders of the project		X	
3	an understanding of the core areas of software engineering		X	
4	an ability to function effectively on teams			X
5	an understanding of professional, ethical and social responsibility			X
6	a recognition of the need for, and an ability to engage in life-long learning	X		
7	knowledge of and ability to apply discrete math, probability and statistics			X
8	the successful design, development, deployment and maintenance of a major software based project		X	
9	an ability to convey technical material through oral presentation and interaction with an audience	X		
10	an ability to convey technical material through written reports which satisfy accepted standards for writing style	X		
11	an ability to evaluate the impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues and emerging software engineering trends, models, tools, and techniques	X		

Associate Degree Assessment Cycle

Table 3-2: Associate Outcome Assessment Timeline

#	Learning Outcomes	10-11	11-12	12-13
1	an ability to identify, formulate, and solve computer programming problems, including the specification, design, implementation, and testing of programs that meet specification, performance, maintenance and quality requirements			X
2	an understanding of the core areas of software engineering		X	
3	an understanding of professional, ethical and social responsibility			X
4	a recognition of the need for, and an ability to engage in life-long learning	X		
5	the successful design and development of a computer program		X	
6	an ability to communicate through oral presentation and interaction with an audience	X		
7	an ability to convey technical material through written reports which satisfy accepted standards for writing style	X		

IV. Summary of Assessment Activities

BS 6 - a recognition of the need for, and an ability to engage in life-long learning

Assessment Activity #1

Course used for assessment: CST 415 – Computer Networks

Instructor/Evaluator: James Long

Student level: Senior

Term of administration: Fall 2010

Number of students: 19

Assessed work: Lifelong Learning Paper

Type of assessment: Direct

To assess this outcome for the institution, students were asked to write a paper given the directions summarized below.

For this assignment, please prepare a paper of approximately 1000 words (four full pages), using a double-spaced format. Your paper should address the four broad areas described below. The bulleted items are suggested topics to help you develop your ideas, but you may add your own ideas as well. *Your paper should be written to form a satisfying whole on the subject of your future career and lifelong learning in your profession, rather than as a series of separate answers to the required areas.* In addition to content (topics below), your paper will also be assessed on purpose, organization, support, style, conventions, and use/citation of outside sources. The attached rubrics (2) will be used to evaluate your paper. Please submit two copies of your paper.

This paper was then used to assess both lifelong learning and written communications. The results of the assessment are shown in Table 4.1.

Table 4-1: BS 6 - Life-long Learning Summary

	Limited or No Proficiency	Some Proficiency	Proficiency	High Proficiency	Proficiency %
Lifelong learning	0 of 19 0.00%	0 of 19 0.00%	15 of 19 78.95%	4 of 19 21.05%	100%
Professional societies	1 of 19 5.26%	4 of 19 21.05%	11 of 19 57.89%	3 of 19 15.79%	73.68%
Credentials	1 of 19 5.26%	6 of 19 31.58%	9 of 19 47.37%	3 of 19 15.79%	63.16%
Continuing education	1 of 19 5.26%	6 of 19 31.58%	11 of 19 57.89%	1 of 19 5.26%	63.15%
Short, long term career plans	2 of 19 10.53%	4 of 19 21.05%	10 of 19 52.63%	3 of 19 15.79%	68.42%
Average Student Total:				14.16	
Average Student Percentage:				70.79%	

AE 4 - a recognition of the need for, and an ability to engage in life-long learning

An assessment activity was planned for spring term but it didn't work well with the measurement tool. This will be reassessed next year with a more appropriate tool and assignment specifications.

BS 9 - an ability to convey technical material through oral presentation and interaction with an audience

Assessment Activity #2

Course used for assessment: CST 412 – Senior Development Project I
Instructor/Evaluator: Calvin Caldwell
Student level: Senior
Term of administration: Fall 2010
Number of students: 25
Assessed work: Project Proposal
Type of assessment: Direct

Students presented their Senior Projects to their classmates. These presentations were the assessed using a predefined institutional rubric. The results of the assessment are shown in Table 4.2.

Table 4-2: BS 9 – Oral presentation summary

	Limited or No Proficiency	Some Proficiency	Proficiency	High Proficiency	Proficiency %
Content	0 of 25 0.00%	0 of 25 0.00%	3 of 25 12.00%	22 of 25 88.00%	100%
Organization	0 of 25 0.00%	0 of 25 0.00%	5 of 25 20.00%	20 of 25 80.00%	100%
Style	0 of 25 0.00%	0 of 25 0.00%	3 of 25 12.00%	22 of 25 88.00%	100%
Delivery	0 of 25 0.00%	1 of 25 4.00%	10 of 25 40.00%	14 of 25 56.00%	96%
Visuals	0 of 25 0.00%	1 of 25 4.00%	2 of 25 8.00%	22 of 25 88.00%	96%
Average Student Total:				18.92	
Average Student Percentage:				94.60%	

AE 6 - an ability to communicate through oral presentation and interaction with an audience

Assessment Activity #3

Course used for assessment: CST 238 – Graphical User Interfaces
Instructor/Evaluator: Randal Albert
Student level: Sophomore
Term of administration: Spring 2011
Number of students: 31
Assessed work: Tip or Trick Presentation
Type of assessment: Direct

One of the assignments for this course was to present a “tip or trick” in regards to some aspect of GUI programming. The results of the assessment are shown in Table 4.3.

Table 4-3: AE 6 – Oral presentation summary

	Limited or No Proficiency	Some Proficiency	Proficiency	High Proficiency	Proficiency %
Content	0 of 31 0.00%	6 of 31 19.35%	24 of 31 77.42%	1 of 31 3.23%	80.65%
Organization	0 of 31 0.00%	6 of 31 19.35%	23 of 31 74.19%	2 of 31 6.45%	80.65%
Style	0 of 31 0.00%	8 of 31 25.81%	20 of 31 64.52%	3 of 31 9.68%	74.19%
Delivery	0 of 31 0.00%	4 of 31 12.90%	18 of 31 58.06%	9 of 31 29.03%	87.10%
Visuals	0 of 31 0.00%	17 of 31 54.84%	11 of 31 35.48%	3 of 31 9.68%	55.16%
Average Student Total:				14.26	
Average Student Percentage:				71.29%	

BS 10 - an ability to convey technical material through written reports which satisfy accepted standards for writing style

Assessment Activity #4

Course used for assessment: CST 415 – Computer Networks
 Instructor/Evaluator: James Long
 Student level: Senior
 Term of administration: Fall 2010
 Number of students: 19
 Assessed work: Lifelong Learning Paper
 Type of assessment: Direct

To assess this outcome for the institution, students were asked to write a paper given the directions summarized below.

For this assignment, please prepare a paper of approximately 1000 words (four full pages), using a double-spaced format. Your paper should address the four broad areas described below. The bulleted items are suggested topics to help you develop your ideas, but you may add your own ideas as well. *Your paper should be written to form a satisfying whole on the subject of your future career and lifelong learning in your profession, rather than as a series of separate answers to the required areas.* In addition to content (topics below), your paper will also be assessed on purpose, organization, support, style, conventions, and use/citation of outside sources. The attached rubrics (2) will be used to evaluate your paper. Please submit two copies of your paper.

This paper was then used to assess both lifelong learning and written communications. The results of the assessment are shown in Table 4.4.

Table 4-4: BS 10 – Written Communications Summary

	Limited or No Proficiency	Some Proficiency	Proficiency	High Proficiency	Proficiency %
Purpose and ideas	0 of 19 0.00%	0 of 19 0.00%	18 of 19 94.74%	1 of 19 5.26%	100%
Organization	0 of 19 0.00%	6 of 19 31.58%	12 of 19 63.16%	1 of 19 5.26%	68.42%
Support	0 of 19 0.00%	7 of 19 36.84%	10 of 19 52.63%	2 of 19 10.53%	63.16%
Style	0 of 19 0.00%	2 of 19 10.53%	16 of 19 84.21%	1 of 19 5.26%	89.47%
Conventions	1 of 19 5.26%	2 of 19 10.53%	14 of 19 73.68%	2 of 19 10.53%	84.21%
Documentation	14 of 19 73.68%	1 of 19 5.26%	2 of 19 10.53%	2 of 19 10.53%	21.06%
Average Student Total:				15.95	
Average Student Percentage:				66.45%	

AE 7 - an ability to convey technical material through written reports which satisfy accepted standards for writing style

Assessment Activity #5

Course used for assessment: CST 223 – Concepts of Programming Languages
Instructor/Evaluator: Sherry Yang
Student level: Sophomore
Term of administration: Spring 2011
Number of students: 16
Assessed work: Term Project Paper
Type of assessment: Direct

Students, either individually or in teams of two, were required to learn and present a new language of their choice, They presented their language to the class as well as writing a paper in regards to their language. The summary of these results are shown in Table 4.5.

Table 4-5: AE 7 – Written Communications Summary

	Limited or No Proficiency	Some Proficiency	Proficiency	High Proficiency	Proficiency %
Purpose and ideas	6 of 16 37.50%	6 of 16 37.50%	3 of 16 18.75%	1 of 16 6.25%	25.00%
Organization	6 of 16 37.50%	6 of 16 37.50%	2 of 16 12.50%	2 of 16 12.50%	25.00%
Support	8 of 16 50.00%	3 of 16 18.75%	3 of 16 18.75%	2 of 16 12.50%	31.25%
Style	6 of 16 37.50%	5 of 16 31.25%	4 of 16 25.00%	1 of 16 6.25%	31.25%
Conventions	5 of 16 31.25%	5 of 16 31.25%	5 of 16 31.25%	1 of 16 6.25%	37.50%
Documentation	6 of 16 37.50%	7 of 16 43.75%	2 of 16 12.50%	1 of 16 6.25%	18.75%
Average Student Total:				11.88	
Average Student Percentage:				49.48%	

BS 11 - an ability to evaluate the impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues and emerging software engineering trends, models, tools, and techniques

Assessment Activity #6

Course used for assessment: ANTH 452 – Globalization
Instructor/Evaluator: Mark Neupert
Student level: Senior
Term of administration: Spring 2011

Number of students: 5
Assessed work: “Globalization is Good” Film
Type of assessment: Direct

Five BSOF students were enrolled in Anthropology 452 during the Spring quarter of 2011. Their progress was assessed using a variety of assignments including classroom discussion, written assignments and video submissions, wherein the student was required to video capture an oral answer to the assignment. The video approach is particularly useful in assessing an individual student’s ability to control the material.

Three areas of learning were assessed.

- 1) The student’s ability to identify and apply appropriate definitions of globalization to examples of global phenomenon. Definitions include liberalization, internationalization, westernization, homogenization, and supraterritoriality. The students were required to identify which definition fit best to reports on current affairs. Example, a free-trade deal between the US and Korea would be best defined as liberalization.
- 2) The student’s ability to apply theory to explain global phenomenon. Theories included neoliberalism, political realism, feminism, post-modernism, Marxism and so on. Students were required to explain global phenomenon from a variety of frameworks. Example, students were required to analyze a film called “Globalization is Good” – identify the framework used by the filmmaker (liberalization) and then asked to reframe, using competing theories, the issues identified in the film (such as explaining and discussing the all-female workforce in a Vietnamese sweatshop in terms of Feminist Theory.)
- 3) Student’s ability to demonstrate control over a variety of current global issues. These issues included the impact of globalization on health, warfare, and food production among others.

The results of the assessment activity are shown in Table 4-6.

	Limited or No Proficiency	Some Proficiency	Proficiency	High Proficiency	Proficiency %
Identification and application of definitions and types of globalization	0 of 5 0%	0 of 5 0%	1 of 5 20.00%	4 of 5 80.00%	100.00%
Identification and application of explanatory frameworks to globalization	0 of 5 0%	0 of 5 0%	3 of 5 60.00%	2 of 5 40.00%	100.00%
Understanding of the impact of globalization on social, economic, and political activities	0 of 5 0%	0 of 5 0%	2 of 5 40.00%	3 of 5 60.00%	100.00%

V. Summary of Student Learning Outcomes

BS 6 - a recognition of the need for, and an ability to engage in life-long learning

Strengths: Students understood the importance of being able to continue learning after they have completed their initial education.

Weaknesses: Although students recognized the need for continual self-learning activities, they were not convinced that advanced education or additional credentials might be necessary.

Action Items: None at this time.

AE 4 - a recognition of the need for, and an ability to engage in life-long learning

Action Items: Redo the assessment activity and tool to make both more appropriate for a sophomore level course.

BS 9 - an ability to convey technical material through oral presentation and interaction with an audience

Strengths: Students were strong in all criteria of this assessment activity.

Weaknesses: None.

Action Items: None at this time.

AE 6 - an ability to communicate through oral presentation and interaction with an audience

Strengths: The students' delivery was very strong. They were able to speak clearly and convey their tips or tricks to their fellow students.

Weaknesses: Students need to understand the role and importance of visual aids in a presentation.

Action Items: None at this time.

BS 10 - an ability to convey technical material through written reports which satisfy accepted standards for writing style

Strengths: Students were able to write clearly and the main ideas presented in the paper were focused.

Weaknesses: The documentation criterion was low across campus. More discussion will take place in the fall to determine an appropriate course of action. The rubric used maybe ambiguous in this specific criteria.

Action Items: Follow up during fall term.

AE 7 - an ability to convey technical material through written reports which satisfy accepted standards for writing style

Strengths: None.

Weaknesses: All, but the documentation criteria was the worst.

Action Items: Follow up during fall term.

BS 11 - an ability to evaluate the impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues and emerging software engineering trends, models, tools, and techniques

Strengths: The five BSOF students demonstrate a strong level of proficiency in the assessed areas. This is a particularly pleasant result in the area of application of social science theory, as these students do not come to this course with a strong background. That these students could accurately apply theoretical frameworks such as Marxism and Neoliberalism to social and economic relations is very encouraging.

Weaknesses: None.

Action Items: None.

VI. Changes Resulting from Assessment

BS 1 - an ability to identify, formulate, and solve software engineering problems, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements

AE 1 - an ability to identify, formulate, and solve computer programming problems, including the specification, design, implementation, and testing of programs that meet specification, performance, maintenance and quality requirements

As a result of assessing these SLOs, more emphasis was made on UML and design techniques in several of our sophomore level classes. The faculty will determine if this was effective during the next assessment cycle.

BS 5 and AE 3- an understanding of professional, ethical and social responsibility

Faculty strengthened their presentation and added an in class exercise on ethical behaviors. The faculty will determine if this was effective during the next assessment cycle.

Appendix A

Course Mapping Matrices

(Note: Courses shaded in red will be used as point to assess the respective SLOs)

BS #6: a recognition of the need for, and an ability to engage in life-long learning

Course	Teach	Eval		
CST 102 – Introduction to Computer Systems	L	L		E- Extensive - a major focus of the course
CST 116 – C++ Programming I				M- Moderate - subject explicitly discussed in and class materials provided
CST 126 – C++ Programming II				L- Little explicit discussion - student may gain the skill due to activities
CST 130 – Computer Organization				
CST 131 – Computer Architecture				
CST 136 – Object Oriented Programming with C++				
CST 162 – Introduction to Digital Logic				
CST 211 – Data Structures		L		
CST 223 - Concepts of Programming Languages				
CST 229 – Grammars				
CST 236 - Software System Testing				
CST 238 – GUI Programming		M		
CST 240 – Unix				
CST 250 – Assembly Language Programming				
CST 276 - Software Design Patterns				
CST 316 – Software Process Management		E		
CST 326 – Software Design and Implementation I		E		
CST 336 – Software Design and Implementation II		E		
CST 320 – Compiler Methods				
CST 324 – Database Systems and Design		L		
CST 334 – Project Proposal				
CST 352 – Operating Systems		M		
CST 412 – Senior Development Project		E		
CST 422 – Senior Development Project		E		
CST 432 – Senior Development Project		E		
CST 415 – Computer Networks		L		

AE #4: a recognition of the need for, and an ability to engage in life-long learning				
Course	Teach	Eval		
CST 102 – Introduction to Computer Systems				E- Extensive - a major focus of the course
CST 116 – C++ Programming I				M- Moderate - subject explicitly discussed in and class materials provided
CST 126 – C++ Programming II				L- Little explicit discussion - student may gain the skill due to activities
CST 130 – Computer Organization				
CST 131 – Computer Architecture				
CST 136 – Object Oriented Programming with C++				
CST 162 – Introduction to Digital Logic				
CST 211 – Data Structures		L		
CST 223 – Concepts of Programming Languages				
CST 236 - Software System Testing				
CST 238 – GUI Programming		M		
CST 240 – Unix				
CST 276 - Software Design Patterns				

BS #9: an ability to convey technical material through oral presentation and interaction with an audience				
Course	Teach	Eval		
CST 102 – Introduction to Computer Systems				E- Extensive - a major focus of the course
CST 116 – C++ Programming I				M- Moderate - subject explicitly discussed in and class materials provided
CST 126 – C++ Programming II				L- Little explicit discussion - student may gain the skill due to activities
CST 130 – Computer Organization				
CST 131 – Computer Architecture				
CST 136 – Object Oriented Programming with C++				
CST 162 – Introduction to Digital Logic				
CST 211 – Data Structures				
CST 223 - Concepts of Programming Languages				
CST 229 – Grammars				
CST 236 - Software System Testing				
CST 238 – GUI Programming	L	L		
CST 240 – Unix				
CST 250 – Assembly Language Programming				
CST 276 - Software Design Patterns				
CST 316 – Software Process Management		L		
CST 326 – Software Design and Implementation I		L		
CST 336 – Software Design and Implementation II		L		
CST 320 – Compiler Methods				
CST 324 – Database Systems and Design				
CST 334 – Project Proposal				
CST 352 – Operating Systems				
CST 412 – Senior Development Project				
CST 422 – Senior Development Project				
CST 432 – Senior Development Project	L	L		
CST 415 – Computer Networks				

AE #6: an ability to convey technical material through oral presentation and interaction with an audience				
Course	Teach	Eval		
CST 102 – Introduction to Computer Systems				E - Extensive - a major focus of the course
CST 116 – C++ Programming I				M - Moderate - subject explicitly discussed in and class materials provided
CST 126 – C++ Programming II				L - Little explicit discussion - student may gain the skill due to activities
CST 130 – Computer Organization				
CST 131 – Computer Architecture				
CST 136 – Object Oriented Programming with C++				
CST 162 – Introduction to Digital Logic				
CST 211 – Data Structures				
CST 223 – Concepts of Programming Languages		L		
CST 236 - Software System Testing				
CST 238 – GUI Programming		L		
CST 240 – Unix				
CST 276 - Software Design Patterns				

BS #10: an ability to convey technical material through written reports which satisfy accepted standards for writing style				
Course	Teach	Eval		
CST 102 – Introduction to Computer Systems	L	L		E- Extensive - a major focus of the course
CST 116 – C++ Programming I				M- Moderate - subject explicitly discussed in and class materials provided
CST 126 – C++ Programming II				L- Little explicit discussion - student may gain the skill due to activities
CST 130 – Computer Organization				
CST 131 – Computer Architecture				
CST 136 – Object Oriented Programming with C++				
CST 162 – Introduction to Digital Logic				
CST 211 – Data Structures				
CST 223 - Concepts of Programming Languages				
CST 229 – Grammars				
CST 236 - Software System Testing				
CST 238 – GUI Programming				
CST 240 – Unix				
CST 250 – Assembly Language Programming				
CST 276 - Software Design Patterns				
CST 316 – Software Process Management		M		
CST 326 – Software Design and Implementation I		M		
CST 336 – Software Design and Implementation II		M		
CST 320 – Compiler Methods				
CST 324 – Database Systems and Design				
CST 334 – Project Proposal	L	E		
CST 352 – Operating Systems				
CST 412 – Senior Development Project				
CST 422 – Senior Development Project				
CST 432 – Senior Development Project				
CST 415 – Computer Networks				

AE #7: an ability to convey technical material through written reports which satisfy accepted standards for writing style				
Course	Teach	Eval		
CST 102 – Introduction to Computer Systems	L	L		E- Extensive - a major focus of the course
CST 116 – C++ Programming I				M- Moderate - subject explicitly discussed in and class materials provided
CST 126 – C++ Programming II				L- Little explicit discussion - student may gain the skill due to activities
CST 130 – Computer Organization				
CST 131 – Computer Architecture				
CST 136 – Object Oriented Programming with C++				
CST 162 – Introduction to Digital Logic				
CST 211 – Data Structures				
CST 223 – Concepts of Programming Languages				
CST 236 - Software System Testing				
CST 238 – GUI Programming				
CST 240 – Unix				
CST 276 - Software Design Patterns				

BS #11: an ability to evaluate the impact of potential solutions to software engineering problems in a global society, using their knowledge of contemporary issues and emerging software engineering trends, models, tools, and techniques				
Course	Teach	Eval	SLO	
CST 102 – Introduction to Computer Systems				E - Extensive - a major focus of the course
CST 116 – C++ Programming I				M - Moderate - subject explicitly discussed in and class materials provided
CST 126 – C++ Programming II				L - Little explicit discussion - student may gain the skill due to activities
CST 130 – Computer Organization				
CST 131 – Computer Architecture				
CST 136 – Object Oriented Programming with C++				
CST 162 – Introduction to Digital Logic				
CST 211 – Data Structures				
CST 223 - Concepts of Programming Languages				
CST 229 – Grammars				
CST 236 - Software System Testing				
CST 238 – GUI Programming				
CST 240 – Unix				
CST 250 – Assembly Language Programming				
CST 276 - Software Design Patterns				
CST 316 – Software Process Management				
CST 326 – Software Design and Implementation I				
CST 336 – Software Design and Implementation II				
CST 320 – Compiler Methods				
CST 324 – Database Systems and Design				
CST 334 – Project Proposal				
CST 352 – Operating Systems				
CST 412 – Senior Development Project				
CST 422 – Senior Development Project				
CST 432 – Senior Development Project				
CST 415 – Computer Networks				
ANTH 452 – Globalization	E	E		

