

B.S., B.A, Minor in Computer Science

Educational Effectiveness

Assessment Plan

Version 1.3

Adopted by

The Computer Science faculty: 6/15/2004

Submitted to

The Dean of the College of Arts & Sciences: 6/15/2004
The Office of Academic Affairs:

TABLE OF CONTENTS

<i>Introduction</i> _____	3
<i>Program Outcomes</i> _____	4
<i>Assessment Tools</i> _____	6
<i>Assessment Implementation & Analysis for Program Improvement</i> _____	8
General Implementation Strategy _____	8
Method of Data Analysis and Formulation of Recommendations for Program Improvement _____	8
Modification of the Assessment Plan _____	9
<i>Appendix A: Faculty Review of Software Development Project</i> _____	10
<i>Appendix B: ETS Major Field Test in Computer Science</i> _____	14
<i>Appendix C: Student Exit Survey</i> _____	20

INTRODUCTION

This document defines the educational outcomes for the Computer Science program and outlines a plan for assessing the achievement of the stated outcomes.

Historically, the assessment strategy in Computer Science was based on determining how prepared and effective our graduates are for prospective employers. Consequently, assessment focused on supervisor and faculty evaluations of our capstone courses, CS A495 (Internship in Computing) and CS A470 (Applied Software Development Project).

While this assessment strategy has been useful, during the 2002-2003 academic year we determined that our strategy was skewed toward software development and project management. The outcomes do not directly address student understanding in several core and theoretical areas of computer science. Additionally, although our program lacks the resources for program accreditation via the Accreditation Board for Engineering and Technology (ABET), we have organized the program to address the accreditation criteria*. Some of these criteria include theoretical areas of computer science. As a result, we modified our assessment plan beginning in the 2003-2004 year to include:

- Two outcomes to address ABET criteria and measure core/theoretical areas of computer science
- Use of Educational Testing Service (ETS) Major Field Test in Computer Science to measure these outcomes
- Exit survey of CS graduates

* http://www.abet.org/criteria_cac.html

PROGRAM OUTCOMES

The overall goal of the Computer Science program is to offer courses covering the major areas of computer science. These courses constitute the basis for an undergraduate major which prepares the student for a variety of professional and technical careers in business, industry and government, or for graduate work leading to advanced degrees.

We have organized our program outcomes based on the 2003-2004 ABET accreditation standards from section IV for Computer Science programs that are listed below:

- IV-1. The curriculum must include at least 40 semester hours of up-to-date study in computer science topics.
- IV-2. The curriculum must contain at least 30 semester hours of study in mathematics and science as specified below under Mathematics and Science.
- IV-3. The curriculum must include at least 30 semester hours of study in humanities, social sciences, arts and other disciplines that serve to broaden the background of the student.
- IV-4. The curriculum must be consistent with the documented objectives of the program.
- IV-5. All students must take a broad-based core of fundamental computer science material consisting of at least 16 semester hours.
- IV-6. The core materials must provide basic coverage of algorithms, data structures, software design, concepts of programming languages, operating systems, and computer organization and architecture.
- IV-7. Theoretical foundations, problem analysis, and solution design must be stressed within the program's core materials.
- IV-8. Students must be exposed to a variety of programming languages and systems and must become proficient in at least one higher-level language.
- IV-9. All students must take at least 16 semester hours of advanced course work in computer science that provides breadth and builds on the core to provide depth.
- IV-10. The curriculum must include at least 15 semester hours of mathematics.
- IV-11. Course work in mathematics must include discrete mathematics, differential and integral calculus, and probability and statistics.
- IV-12. The curriculum must include at least 12 semester hours of science.
- IV-13. Course work in science must include the equivalent of a two-semester sequence in a laboratory science for science or engineering majors.
- IV-14. Science course work additional to that specified in Standard IV-13 must be in science courses or courses that enhance the student's ability to apply the scientific method.
- IV-15. The oral communications skills of the student must be developed and applied in the program.
- IV-16. The written communications skills of the student must be developed and applied in the program.
- IV-17. There must be sufficient coverage of social and ethical implications of computing to give students an understanding of a broad range of issues in this area.

The UAA Computer Science Program has adopted many of these standards within our curriculum and they have influenced us to adopt the following educational outcomes:

- 1. Students should develop oral communications skills consistent with a career in CS.**
- 2. Students should develop written communication skills consistent with a career in CS.**
- 3. Students should demonstrate abilities in critical thinking, problem solving and analysis skills, and software design.**
- 4. Students should demonstrate abilities in software development and implementation.**
- 5. Students should demonstrate basic coverage of core concepts in CS, including algorithms, data structures, concepts of programming languages, operating systems, and computer organization and architecture.**
- 6. Students should demonstrate basic understanding of theoretical foundations of CS including discrete mathematics, algorithm analysis, and computability.**

Outcomes 1-4 were implemented prior to 2003-2004. Outcomes 5 and 6 are adapted from the ABET accreditation standards and are in use beginning with the 2003-2004 academic year.

ASSESSMENT TOOLS

A description of the tools used in the assessment of the program outcomes and their implementation are summarized in Table 1. The tools and their relationships to the program objectives are listed in Table 2.

Note that the use of the Exit Survey and ETS Major Field Test did not begin until Spring 2004.

There is a separate appendix for each tool that describes the factors that affect the results and give examples of the tools and how they will be implemented.

Table 1
Program Outcomes Assessment Tools and Administration

Tool	Description	Frequency/ Start Date	Collection Method	Administered by
Capstone Course	Capstone coursework is evaluated by panel of CS faculty for outcomes 1-2 and by the primary supervisor for outcomes 3-4.	Yearly / Fall 2000	Faculty Review	All CS Faculty
Exit Survey	Graduating students are asked to directly provide feedback on the effectiveness of the entire program.	Yearly / Spring 2004	Mailed survey	CS A470 Instructor
ETS Major Field Test	Worldwide standardized exam for undergraduates used to measure student academic achievement and growth.	Yearly / Spring 2004	Administered in CS A470 course	CS A470 Instructor

Table 2
Association of Assessment Tools to Program Objectives

	Capstone Course	Exit Survey	ETS Major Field Test
1. Students should develop oral communications skills consistent with a career in CS.	1	1	0
2. Students should develop written communication skills consistent with a career in CS.	1	1	0
3. Students should demonstrate abilities in critical thinking, problem solving and analysis skills, and software design.	1	1	0
4. Students should demonstrate abilities in software development and implementation.	1	1	1
5. Students should demonstrate basic coverage of core concepts in CS, including algorithms, data structures, concepts of programming languages, operating systems, and computer organization and architecture.	0	1	1
6. Students should demonstrate basic understanding of theoretical foundations of CS including discrete mathematics, algorithm analysis, and computability.	0	1	1

0 = Tool is not used to measure the associated objective.
1 = Tool is used to measure the associated objective.

ASSESSMENT IMPLEMENTATION & ANALYSIS FOR PROGRAM IMPROVEMENT

General Implementation Strategy

Implementation of our assessment plan revolves around CS A470 and CS A495. One of these courses is required by all BA and BS students in the program. CS A470 is the Applied Software Development Project while CS A495 is reserved for an internship of equal complexity to a CS A470 project. CS A470 requires the student to propose, design, implement, and document a project of moderate to large complexity. We believe that this capstone course is a good indicator of a student's performance since a successful project requires the integration of knowledge from many other courses. Programming skills, algorithm development, software design, database issues, writing skills, and presentation skills are all required elements of the course.

The course projects are used to directly measure outcomes 1-4 (written, oral presentation, critical thinking, and software implementation skills). A committee of preferably three or more faculty members performs the evaluation on outcomes 1-2 for each student. By using a committee, we avoid biases that may result if a single faculty member performs the assessment. Each faculty member grades the student from a scale of A-F and also includes specific comments for that student (e.g. specific areas of improvement or lessons learned). Either the CS A470 instructor or a supervising faculty member performs the assessment on outcomes 3-4 for each student.

Faculty review of the student projects is a fairly long and time-consuming process. To assist the assessment process, a website was established where CS faculty can read student reports, view oral presentations, view source code, and enter assessment grades online. By moving the process online, bulky paper reports no longer need to be circulated among faculty and faculty are able to view and enter assessment grades from any location with a web browser.

Since Spring of 2004, CS A470 and CS A495 are also the focal points for measuring outcomes 5-6 (core and theoretical CS skills) through the ETS Major Field Test in Computer Science. Additionally, upon conclusion of the CS A470 course students are given an exit survey to complete and mail in so that they may provide direct feedback on all outcomes.

Data is to be collected primarily by the CS A470 instructor with assistance from the department staff to collect exit surveys.

Method of Data Analysis and Formulation of Recommendations for Program Improvement

The faculty of the program is to meet at least once a year to review the data collected using the assessment tools. The primary analyst of the data is expected to be the CS A470 instructor. This meeting should result in recommendations for program changes that are designed to enhance performance relative to the program's objectives and outcomes. The results of the data collection, an interpretation of the results, and the recommended programmatic changes are to be forwarded to the office of Academic Affairs (in the required format) by June 15 each year. A plan for implementing the

recommended changes, including of advertising the changes to all the program's stakeholders, is also to be completed at this meeting.

The proposed programmatic changes may be any action or change in policy that the faculty deems as being necessary to improve performance relative to programs objectives and outcomes. Recommended changes should also consider workload (faculty, staff, and students), budgetary, facilities, and other relevant constraints. A few examples of changes made by programs at UAA include:

- changes in course content, scheduling, sequencing, prerequisites, delivery methods, etc.
- changes in faculty/staff assignments
- changes in advising methods and requirements
- addition and/or replacement of equipment
- changes to facilities

Modification of the Assessment Plan

The faculty, after reviewing the collected data and the processes used to collect it, may decide to alter the assessment plan. Changes may be made to any component of the plan, including the objectives, outcomes, assessment tools, or any other aspect of the plan. The changes are to be approved by the faculty of the program. The modified assessment plan is to be forwarded to the dean/director's office and the Office of Academic Affairs.

APPENDIX A: FACULTY REVIEW OF SOFTWARE DEVELOPMENT PROJECT

Tool Description:

The software development project, CS A470, is a capstone course intended to provide the students with the opportunity to apply their education to the planning and design of a software system of moderate to large complexity. The project is chosen by the student in combination with the faculty. Students are encouraged to obtain a “real” client with a project that involves as many of the CS disciplines as possible. The class is organized so that students deliver a project proposal, specifications, design document, code review, implementation, demonstration, project presentation, and final report throughout the semester.

A committee of CS faculty evaluates each project based upon the oral presentation, project demo, final report, and source code in terms of outcomes 1-2. The CS A470 instructor or supervising faculty member assesses outcomes 3-4.

Evaluation is input via a website available to faculty. Figure 1 shows an example of the main entry to the site. By clicking on the link in ‘Documents’ a reviewer may view the project report, presentation, and source code. Figure 2 depicts the process of entering a review for outcomes 1-4 and figure 3 shows all reviews entered for a student.

Figure 1.
Website Sample: Main entry to web site assessment tool

Name	Documents	Assessment
John Doe	Online Faculty Workload and Activity Report System	View all assessments Edit your assessment
Susan Doe	Online Job Posting System	View all assessments Edit your assessment

Figure 2.
Website Sample: Entering a review

Student: **John Doe**
Reviewer: **Faculty 1**

<p>Students should develop oral communications skills consistent with a career in CS.</p>	<p>Grade (A-F): <input type="text" value="B+"/></p> <p>Comments: <input type="text" value="Slides looked nice, the student spoke relat"/></p>
<p>Students should develop written communications skills consistent with a career in CS.</p>	<p>Grade (A-F): <input type="text" value="B+"/></p> <p>Comments: <input type="text" value="Could use more proofreading. For exampl"/></p>
<p>Students should demonstrate critical thinking and problem solving skills in CS.</p>	<p>Grade (A-F): <input type="text" value="A"/></p> <p>Comments: <input type="text" value="The students learned a great deal on their"/></p>
<p>Students should demonstrate abilities in problem analysis and software design, development, and implementation.</p>	<p>Grade (A-F): <input type="text" value="A-"/></p> <p>Comments: <input type="text" value="The students would have benefited a grea"/></p>

Figure 3.
Website Sample: Viewing All Reviews

Reviewer: **Faculty 1**

Subject	Grade	Comments
Students should develop oral communications skills consistent with a career in CS.	A	The student spoke clearly and gave a presentation at a level to be expected for CS 470/495.
Students should develop written communications skills consistent with a career in CS.	A	The written report was of an appropriate length and depth for a project of this sort. Please see additional, general comments under point 4.
Students should demonstrate critical thinking and problem solving skills in CS.	A	The project was based on a real-world problem and it was evident that the students encountered the kinds of challenges that arise in such environments.
Students should demonstrate abilities in problem analysis and software design, development, and implementation.	A	Based on the presentation it was clear that the students had dealt with the elements of this category. In the oral presentation, in the powerpoint slides, and in the written report, more detail might have been helpful. I am thinking specifically of sections 4.2, 4.3, and 4.4 of the final report.

Reviewer: **Faculty 2**

Subject	Grade	Comments
Students should develop oral communications skills consistent with a career in CS.	A	The presenter seemed to be comfortable and did a good job working as a team. Dressing up for the presentation shows that they care.
Students should develop written communications skills consistent with a career in CS.	A	The report is generally well written. I like that they were open and honest in some of the problems they had during development. I think they whined too much about the clients and lack of guidelines.
Students should demonstrate critical thinking and problem solving skills in CS.	B	The biggest challenge to this project appears to be its size and the changing needs of the clients. While they became frustrated with that issue, their response to the problem appears to be blaming the client for not producing the specifications and requirements for them rather than assuming that as part of their job. The main problem with this project is that it seems destined to never be implemented.
Students should demonstrate abilities in problem analysis and software design, development, and implementation.	B	The student was too busy dealing with changing user requirements to do any serious abstractions that could make extending the project easy. By their own admission they organized the database for ease of use (at the time) rather than on a long term design. Even though the developers lament the fact that the ultimate desires of the clients are not clear, they should still be able to devise an architecture that allows them to develop a web application that allows for future extensions and improvements. Had they done some sort of modular decomposition and reviewed their design an architecture on a regular basis, they probably would have had an easier time dealing with changing desires.

Factors that affect the collected data:

- Time and energy requirements. Bias associated with the grading philosophy of a single faculty member is avoided by using a committee of reviewers. However, this requires all faculty to invest significant time and effort outside of their normal workload to assess each project.
- Grading philosophy of faculty. Although a committee is used to avoid the grading bias of a single faculty member for outcomes 1-2, the committee size is small. With only 3-4 faculty, an individual's grading bias still has a relatively strong impact on the final assessment of grades. Additionally, a single faculty member is responsible to assess outcomes 3-4. However, the grading philosophy on these outcomes may be tempered by overlap with the ETS Major Field Test examination (appendix B).

How to interpret the data:

The data directly corresponds to how well faculty perceives students are performing in relation to the first four outcomes.

Tabulating and Reporting Results

The comments and average results are used to examine potential modifications to the assessment plan. The faculty assessment grades are averaged for all students across outcomes 1-4 using the grading scale of:

A = 4.0, A- = 3.7 B+ = 3.3, B = 3.0 B- = 2.7
C+ = 2.7 C = 2.0 C- = 1.7
D+ = 1.3 D = 1.0 D- = 0.7 F = 0

The results are tabulated as shown in the example below:

Learning Outcome	Average Faculty Evaluation
1. Students should develop oral communications skills consistent with a career in CS.	3.33
2. Students should develop written communications skills consistent with a career in CS.	3.40
3. Students should demonstrate abilities in critical thinking, problem solving and analysis skills, and software design.	3.94
4. Students should demonstrate abilities in software development and implementation.	3.91

APPENDIX B: ETS MAJOR FIELD TEST IN COMPUTER SCIENCE

Tool Description:

The content of the ETS Major Field Test in Computer Science reflects the basic knowledge and understanding gained in the core undergraduate curriculum. The tests are conducted worldwide and are two-hour, multiple-choice examinations designed to assess mastery of concepts and principles as well as knowledge expected of students at the conclusion of a major in CS. They go beyond measurement of factual knowledge, however, because they also evaluate students' ability to analyze and solve problems, understand relationships, and interpret material.

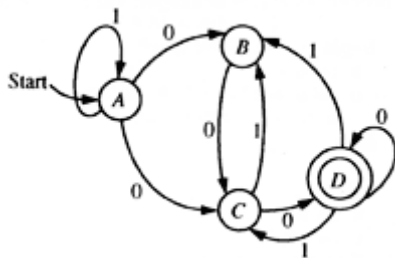
Each test delivers an individual student score report, plus the mean scale score and standard deviation for the group of students tested. The CS test also delivers subscores that can be used to highlight students' strengths or weaknesses in these areas. Additionally, the CS test also delivers assessment indicators relating to the performance of the group of students within subareas of computer science. The Major Field Tests only score correct answers, thereby not penalizing students for any omissions or guesses.

The CS faculty will administer the test during the CS A470 course. The test results will be used to assess outcomes 5-6 and potentially 3-4.

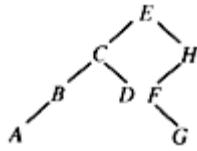
Sample questions from the ETS Major Field Test in Computer Science are shown below.

Major Field Test in Computer Science - Sample Questions

The following questions illustrate the range of the test in terms of the abilities measured, the disciplines covered, and the difficulty of the questions posed. They should not, however, be considered representative of the entire scope of the test in either content or difficulty. An answer key follows the questions.



1. If D is the accepting state of the nondeterministic finite automaton above, which of the following does the automaton accept?
 - A. 001
 - B. 1101
 - C. 01100
 - D. 000110
 - E. 100100



2. If a node in the binary search tree above is to be located by binary tree search, what is the expected number of comparisons required to locate one of the items (nodes) in the tree chosen at random?

- A. 1.75
- B. 2
- C. 2.75
- D. 3
- E. 3.25

Questions 3 and 4 are based on the following information.

If the variables are suitably initialized, and if i remains within appropriate bounds, then the following code implements the stack operations *Push* and *Pop* when the stack is represented as an array $V[1..N]$ with an index variable i .

Push: **begin** $V[i] := x; i := i + 1$ **end**

Pop: **begin** $i := i - 1; x := V[i]$ **end**

3. Which of the following gives the correct initialization for this stack implementation?
- A. $i := 0$
 - B. $i := 1$
 - C. $i := N - 1$
 - D. $i := N$
 - E. $i := N/2$
4. If it is assumed that suitable changes in the initialization code were also made, which of the following changes to *Push* and *Pop* would yield a correct implementation of stacks?
- a. Replacing the code for *Push* with that for *Pop* and vice versa
 - b. Making *Push* decrement i and *Pop* increment i
 - c. Reversing the order of the statements in both *Push* and *Pop*
- A. I only
 - B. II only
 - C. III only
 - D. I and II only
 - E. II and III only

5. In a computer with a cache memory interposed between the processor and the primary memory, the cache is k -way set-associative (for some fixed k); i.e., each location in primary memory "maps to" (can be cached in) any of k locations in the cache. Let there be P locations in primary memory and C locations in the cache.

On the average, how many different locations in primary memory map to a particular location in the cache if $k = 2$?

- A. 1
- B. 2
- C. P/C
- D. $2 P/C$
- E. P

6. Which of the following regular expressions generate(s) no string with two consecutive 1's (Note that ϵ denotes the empty string.)

- I. $(1 + \epsilon)(01 + 0)^*$
- II. $(01 + 10)^*$
- III. $(0 + 1)^*(0 + \epsilon)$

- A. I only
- B. II only
- C. III only
- D. I and II only
- E. II and III only

7. Which of the following C++ expressions does NOT always correctly compute the mathematical average of the integer variables a , b , c , and d ?

- A. `float ((a + b + c + d) / 4.0)`
- B. `(float (a + b + c + d)) / 4`
- C. `(a + b + c + d) / 4`
- D. `(a + b + c + d) / 4.0`
- E. `(a + float (b) + c + d) / 4`

8. Consider the following recursive function.

```
int Fun(int n)
{
    if (n == 4)
        return 2;
    else
        return 2*Fun(n+1);
}
```

What is the value returned by the function call Fun(2) ?

- A. 2
- B. 4
- C. 8
- D. 16
- E. 24

9. If A is an array with n elements and procedure Swap exchanges its arguments, then the following code segment sorts A in descending order.

```
for (int j = 0; j < n-1; j++)
    for (int k = 0; k < n-j-1; k++)
        if (A[k] < A[k+1]) Swap(A[k],A[k+1]);
```

How many calls to Swap are made if initially, $A[i]=i$, for $i = 0, 1, 2, \dots, n-1$?

- A. $n-1$
- B. n
- C. $n(n-1)/2$
- D. $(n-1)(n-2)$
- E. $n(n-1)$

10. Which of the following statements about static RAM (SRAM) and/or dynamic RAM (DRAM) is true?

- A. SRAM is implemented using transistors and capacitors that must be periodically refreshed.
- B. DRAM has a faster access time than SRAM.
- C. DRAM is less expensive than SRAM.
- D. SRAM is capable of operating at speeds closely approximating processors.
- E. DRAM is capable of operating at speeds closely approximating processors.

11. Suppose that $\{S_1, S_2, \dots, S_N\}$ is a set of N fixed-length strings ordered alphabetically so that $S_1 < \dots < S_N$. If these strings are held, in order, in an array of N elements, then the time required to find the location of S_{i+1} , given the location of S_i , $1 \leq i < N$ has order

- A. 1
- B. $\log_2 N$
- C. N
- D. $N \log_2 N$
- E. N^2

12. Suppose that V is a vector with indices from a to b and that each element of V occupies two words. If the elements of V are stored in consecutive words of memory and $V[a]$ is the address of word 1 of $V[a]$, then the address of word 1 of $V[i]$, where $a \leq i \leq b$, is

- A. $(\&V[a] - a) + i$
- B. $2(\&V[a] - a) + 2i$
- C. $(\&V[a] - 2a) + 2i$
- D. $(\&V[a] - 2a) + i$
- E. $(\&V[a] - a) + 2i$

13. If A, B, C are Boolean variables, and if " \wedge " and " \vee " denote Boolean "and" and "or," respectively, which of the following is (are) true?

- I. $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$
- II. $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
- III. $(B \wedge A) \vee C = C \vee (A \wedge B)$

- A. I only
- B. II only
- C. I and II only
- D. II and III only
- E. I, II, and III

Answer Key			
1. E	5. D	9. C	13. E
2. C	6. A	10. C	
3. B	7. C	11. A	
4. E	8. C	12. C	

Factors that affect the collected data:

- Student motivation. The faculty currently has no plans to tie the test scores with course grades, thereby raising the possibility that unmotivated students will not try their best on the exam. Students do have some motivation in seeing where they stand in relation to other CS students nationwide and learning of any potential deficiencies in their studies.
- Cost. With no budget to purchase the exams within the department funding may need to be secured from other sources. We do not believe it is reasonable to require students to pay for their own exams, which is estimated at \$30 each. CAS paid for the tests during 2003-2004 but no funding has been secured for future years.

How to interpret the data:

The ETS reports scores on a scale of 120-200. Scores are provided for each individual student together with subscores ranging from 0-100 in the specific areas of Programming Fundamentals (Outcome 4), Computer Organization/Architecture/Operating Systems (Outcome 5), and Algorithms/Theory/Computation/Math (Outcome 6). Averages are provided for all universities nationwide.

Tabulating and Reporting Results

The test results will be received once a year. The results are combined and scores for the relevant outcomes are computed by faculty together with other assessment data. The results will be scaled to the 0-4 data range (i.e. F to A) for the purpose of analysis.

APPENDIX C: STUDENT EXIT SURVEY

Tool Description:

The exit survey asks graduates of the program to rate their performance relative to the program's outcomes. Additionally, graduates are asked to rate the program's delivery of the material related to the objectives from their viewpoint.

Surveys are distributed to students upon completion of the CS A470 course. The students return the surveys by mail or to the staff. The results are not examined until after grades have been submitted.

A sample of the survey instrument is in the following pages.

Factors that affect the collected data:

A number of factors need to be taken into consideration when analyzing the data. The following factors are those that we have identified.

- Low return rates. It has proven difficult get a good return rate from the graduates. This reduces the accuracy of the results.
- Student knowledge. Students that enroll in CS A470 may not actually graduate until the following year and may not have the knowledge to answer the survey accurately.
- Student effort. The amount of effort students take to accurately complete the survey is variable.

How to interpret the data:

Care should be taken to investigate and discuss the factors influencing the results before interpreting the results. The results of the surveys should also be compared against the other indicators to get good picture of program performance relative the expected outcomes.

Tabulating and Reporting Results

The survey is prepared by the faculty. Staff receives the results and faculty tabulates them for use in outcomes review.

College of Arts & Sciences
Computer Science Program
Graduate Exit Survey

The Computer Science program has received mandates from the university to implement an outcomes-based assessment program. As a part of the program, we are surveying graduating students to find ways of improving our program. Your feedback will go a long way in helping us determine how well we are doing and what we can do to better serve our students, alumni, and the engineering community. Please return in the provided envelope. These surveys will be opened after final grades have been submitted for the semester.

Do you intend to pursue Graduate Study? Yes No

Have you accepted a permanent computer science position? Yes No

Primary area of Computer Science that you hope to work in:

- | | | |
|---|--|--|
| <input type="checkbox"/> Software Development | <input type="checkbox"/> Networking or System Administration | <input type="checkbox"/> Research |
| <input type="checkbox"/> Management | <input type="checkbox"/> Technical Support | <input type="checkbox"/> Database Systems |
| <input type="checkbox"/> Software Testing | <input type="checkbox"/> Systems Analyst | <input type="checkbox"/> Not working in Computer Science |
| <input type="checkbox"/> Other : _____ | | |

The UAA Computer Science program has adopted 6 expected outcomes, please rate your knowledge/skills and the program's effectiveness in teaching you knowledge/skills relative each objective.

1) Students should develop oral communications skills consistent with a career in Computer Science;

What is your understanding/ability now? poor, fair, good, excellent, outstanding, No opinion

How well did we do teaching this? poor, fair, good, excellent, outstanding, No opinion

2) Students should develop written communication skills consistent with a career in Computer Science;

What is your understanding/ability now? poor, fair, good, excellent, outstanding, No opinion

How well did we do teaching this? poor, fair, good, excellent, outstanding, No opinion

3) Students should demonstrate abilities in critical thinking, problem solving and analysis skills, and software design;

What is your understanding/ability now? poor, fair, good, excellent, outstanding, No opinion

How well did we do teaching this? poor, fair, good, excellent, outstanding, No opinion

4) Students should demonstrate abilities in software development and implementation;

What is your understanding/ability now? poor, fair, good, excellent, outstanding, No opinion

How well did we do teaching this? poor, fair, good, excellent, outstanding, No opinion

5) Students should demonstrate basic coverage of core concepts in Computer Science, including algorithms, data structures, concepts of programming languages, operating systems, and computer organization and architecture;

What is your understanding/ability now? poor, fair, good, excellent, outstanding, No opinion

How well did we do teaching this? poor, fair, good, excellent, outstanding, No opinion

6) Students should demonstrate basic understanding of theoretical foundations of Computer Science including discrete mathematics, algorithm analysis, and computability;

What is your understanding/ability now? poor, fair, good, excellent, outstanding, No opinion

How well did we do teaching this? poor, fair, good, excellent, outstanding, No opinion

Please indicate your satisfaction with each of the following aspects of your experience at UAA. Please feel free to use the space after the list to briefly explain any of your responses, especially if you feel less than satisfied.

Quality of the Advising: poor, fair, good, excellent, outstanding, No opinion

Quality of Instruction: poor, fair, good, excellent, outstanding, No opinion

Quality of Physical Facilities: poor, fair, good, excellent, outstanding, No opinion

Quality of Computer Laboratories: poor, fair, good, excellent, outstanding, No opinion

Optional Explanations:

Please list up to three major strengths of your undergraduate computer science education or other UAA experiences.

Please list up to three areas for improvement in our undergraduate computer science program or other aspects of UAA.

With respect to the previous question, do you have any suggestions on how UAA could address these improvements?

Would you recommend a UAA computer science education to a friend or relative? Yes No Maybe