**Bachelor of Science and Bachelor of Arts in Computer Science**

# Academic Assessment Plan

**Version 3.1**

**Adopted by**

**The Computer Science & Engineering faculty: 12/15/2017**

**Approved by CoEng Assessment Committee: 1/16/2018**

Reviewed with curriculum changes by the Academic Assessment Committee as an information item: 2/19/21

Reviewed by the Faculty Senate as an information item: 3/5/21

Reviewed with curriculum changes by the Academic Assessment Committee as an information item: 2/7/20

Reviewed by the Faculty Senate as an information item: 3/6/20

Reviewed by the Academic Assessment Committee: 4/20/18

Reviewed by the Faculty Senate as an information item: 5/4/18

**Table of Contents**

[Mission Statement 3](#_Toc500971932)

[Program Introduction 3](#_Toc500971933)

[Assessment Process Introduction 3](#_Toc500971934)

[Program Student Learning Outcomes 4](#_Toc500971935)

[Mapping Previous Program Student Learning Outcomes to Current Program Student Learning Outcomes 5](#_Toc500971936)

[Table 1: Mapping of Prior To Current Program Student Learning Outcomes 5](#_Toc500971937)

[Table 2: Association of Assessment Measures to Program Student Learning Outcomes 7](#_Toc500971938)

[Assessment Measures 7](#_Toc500971939)

[Table 3: Program Outcomes Assessment Measures and Administration 8](#_Toc500971940)

[Assessment Implementation & Analysis for Program Improvement 9](#_Toc500971941)

[General Implementation Strategy 9](#_Toc500971942)

[Method of Data Analysis and Formulation of Recommendations for Program Improvement 9](#_Toc500971943)

[Modification of the Assessment Plan 11](#_Toc500971944)

[Appendix A: Faculty Review of Student Artifacts 12](#_Toc500971945)

[Appendix B: ETS Major Field Test In Computer Science 16](#_Toc500971946)

[Appendix C: Student Exit Survey 18](#_Toc500971947)

Mission Statement

The Computer Science program at the University of Alaska Anchorage is committed to the University’s mission to discover and disseminate knowledge through teaching, research, engagement, and creative expression.  First, through undergraduate and graduate programs, we strive to teach our students the fundamental principles of computer science and important issues in computing so they may pursue advanced degrees or enter the workplace as productive, competent software development or information technology professionals.   Second, the program seeks to further the profession of computer science through professional activities and public service within the community, state, nation, and society at large.   Finally, the program engages in and disseminates research to advance the development of computer science and provide innovative technological solutions to address the needs of modern society.

Program Introduction

The Computer Science program is housed in the Department of Computer Science & Engineering in the College of Engineering. A common core curriculum consisting of computer programming, computer organization, and networking is shared by the BA/BS in Computer Science and the Bachelor of Science in Computer Systems Engineering.

The BS program is accredited by the Computing Accreditation Commission of ABET, Inc., http://www.abet.org. The BA program does not include the math and science requirements necessary for accreditation through ABET but the computer science content in the BA and BS degrees are identical.

Assessment Process Introduction

On 12/15/2017 the department revised the Program Student Learning Outcomes (PSLOs) to match the CAC 2018 -2019 Criteria Version 2.0. There are now six as opposed to eleven PSLOs.

In Spring of 2013 the department adopted new PSLO’s as part of our efforts to obtain ABET accreditation. The eleven adopted PSLOs closely matched the criteria required by ABET prior to Version 2.0. The requirements included communication skills, ethics, programming skill, and theoretical foundations of Computer Science. Previously, the BA/BS program had six outcomes with only an indirect mapping to the required ABET outcomes. Since the outcomes changed significantly our data analysis started over effective Spring 2013.

The core of the assessment plan is now based on faculty evaluation via rubrics. Student artifacts that are collected in different courses throughout the curriculum are evaluated by faculty members and the results are aggregated in the assessment report.

Currently, our assessment plan includes:

* Educational Testing Service (ETS) Major Field Test in Computer Science
* Evaluation of student performance in CSCE A351, CSCE A401, CSCE A465, and CSCE A470.
* Exit survey of CS graduates.

Program Student Learning Outcomes

At the completion of this program, students will be able to:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.
3. Communicate effectively in a variety of professional contexts, including technical and non-technical audiences for business, end-user, client, and computing contexts.
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

The highlighted text is not an ABET PSLO. It was added to our PSLO’s after discussion from our Advisory Board.

Mapping Previous Program Student Learning Outcomes to Current Program Student Learning Outcomes

The previous PSLOs are listed below. Table 1 illustrates how the previous PSLOs map to the current PSLO’s.

1. Apply knowledge of computing and mathematics appropriate to the discipline.
2. Analyze a problem and identify and define the computing requirements appropriate to its solution.
3. Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
4. Function effectively on teams to accomplish a common goal.
5. Demonstrate an understanding of professional, ethical, legal, security and social issues and responsibilities.
6. Communicate effectively with a range of audiences, including technical and non-technical audiences for business, end-user, client and computing contexts.
7. Analyze the local and global impact of computing on individuals, organizations and society.
8. Recognize the need for and an ability to engage in continuing professional development.
9. Use current techniques, skills, and tools necessary for computing practice.
10. Apply mathematical foundations, algorithmic principles and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
11. Apply design and development principles in the construction of software systems of varying complexity.

Table 1: Mapping of Prior To Current Program Student Learning Outcomes

| **Prior PSLO** | **Current PSLO** |
| --- | --- |
| 2. Analyze a problem and identify and define the computing requirements appropriate to its solution. | 1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. |
| 3. Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.  9. Use current techniques, skills, and tools necessary for computing practice. | 2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline. |
| 6. Communicate effectively with a range of audiences, including technical and non-technical audiences for business, end-user, client and computing contexts. | 3. Communicate effectively in a variety of professional contexts, including technical and non-technical audiences for business, end-user, client, and computing contexts. |
| 5. Demonstrate an understanding of professional, ethical, legal, security and social issues and responsibilities.  7. Analyze the local and global impact of computing on individuals, organizations and society.  8. Recognize the need for and an ability to engage in continuing professional development. | 4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. |
| 4. Function effectively on teams to accomplish a common goal. | 5. Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline. |
| 1. Apply knowledge of computing and mathematics appropriate to the discipline.  10. Apply mathematical foundations, algorithmic principles and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.  11. Apply design and development principles in the construction of software systems of varying complexity. | 6. Apply computer science theory and software development fundamentals to produce computing-based solutions. |

Table 2: Association of Assessment Measures to Program Student Learning Outcomes

| **Outcomes** | CSCE A351 | CSCE A401 | CSCE A465 | CSCE A470 | Exit Survey | ETS Field Test |
| --- | --- | --- | --- | --- | --- | --- |
| 1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. | 0 | 1 | 0 | 1 | 1 | 0 |
| 2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline. | 0 | 1 | 0 | 1 | 1 | 0 |
| 3. Communicate effectively in a variety of professional contexts, including technical and non-technical audiences for business, end-user, client, and computing contexts. | 0 | 1 | 0 | 1 | 1 | 0 |
| 4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles. | 0 | 0 | 1 | 0 | 1 | 0 |
| 5. Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline. | 0 | 1 | 0 | 1 | 1 | 0 |
| 6. Apply computer science theory and software development fundamentals to produce computing-based solutions. | 1 | 0 | 0 | 0 | 1 | 1 |

0 = Measure is not used to measure the associated outcome.

1 = Measure is used to measure the associated outcome.

Assessment Measures

A description of the tools used in the assessment of the program outcomes and their implementation are summarized in Table 2. The tools and their relationships to the program outcomes are listed in Table 3.

Table 3: Program Outcomes Assessment Measures and Administration

| **Measure** | **Description** | **Frequency/ Start Date** | **Collection Method** | **Administered by** |
| --- | --- | --- | --- | --- |
| Evaluation of student coursework | A rubric is used to evaluate student coursework in the context of a particular outcome. Currently work is used from:  CSCE A351, CSCE A465, CSCE A401, and CSCE A470. | Yearly or every semester (depending on course offering) / Spring 2018 | Faculty collection | Instructor of course |
| Exit Survey | Graduating students are asked to directly provide feedback on the effectiveness of the entire program. | Yearly / Spring 2013 | Online survey sent to students in CSCE A470 | Assessment Coordinator |
| ETS Major Field Test | Worldwide standardized exam for undergraduates used to measure student academic achievement and growth. | Yearly / Spring 2004 | Administered in CSCE A470 course | Assessment Coordinator |

Assessment Implementation & Analysis for Program Improvement

General Implementation Strategy

Implementation of our assessment plan revolves around faculty evaluation of student work. Faculty members in selected courses will choose student artifacts (e.g. papers, assignments, presentations) that relate to a particular outcome and will evaluate the work based on a rubric. Evaluation is independent of the grade assigned for the course. In some cases multiple faculty members may evaluate the same outcome. For example, faculty members attending a presentation may all evaluate a student’s ability to communicate effectively. The scores from all evaluators are averaged together in the final analysis.

Two courses in particular are key to our assessment efforts. CSCE A401, Software Engineering, requires students to work in groups, work with a client to elicit and state requirements, and to implement a software project. All of these activities map to PSLOs. In addition, the capstone course, CSCE A470, also requires students to research, specify, design, and implement a project of moderate complexity. This course touches upon most PSLOs. In the CSCE A470 class we also administer the ETS Major Field Test in Computer Science. This test is taken nationally and allows us to compare our students’ performance with other programs nationally and internationally.

Method of Data Analysis and Formulation of Recommendations for Program Improvement

At the end of the spring semester we will collect and aggregate data that was collected in the spring and the previous fall. The data is simply averaged as we move up to higher levels of analysis. However, we will retain the low-level data if we need to drill down to see specific sub-outcomes that may need to be addressed.

For example, consider the following subset of the rubric for PSLO #3:

**Outcome 3:**  **Communicate effectively in a variety of professional contexts, including technical and non-technical audiences for business, end-user, client, and computing contexts.**

| **Evaluation Criteria** | **Poor** | **Developing** | **Satisfactory** | **Excellent** |
| --- | --- | --- | --- | --- |
| 1. Effectively organizes and structures a presentation or document | No logical structure | Some structure but erratic jumps in topic | Most information presented logically | All information presented logically |
| 1. Provides appropriate content to demonstrate detailed knowledge of subject area | No grasp of topic, cannot answer questions or extremely limited content | Only rudimentary knowledge demonstrated | At ease with content and provides some detail | Full command of subject matter |

Student #1 receives a score of “Poor” for Outcome 3.1, and a score of “Satisfactory” for Outcome 3.2

Student #2 receives a score of “Satisfactory” for Outcome 3.1 and a score of “Excellent” for Outcome 3.2.

These scores would be aggregated as percentages for each sub-outcome. The average of the percentages is then computed for the outcome overall. Our simple example with students 1 and 2 results in the following scores:

|  | Poor | Developing | Satisfactory | Excellent |
| --- | --- | --- | --- | --- |
| Outcome 3.1 | 50% (1/2) | 0% | 50% (1/2) | 0% |
| Outcome 3.2 | 0% | 0% | 50% (1/2) | 50% (1/2) |
| Outcome 3 Overall | 25% | 0% | 50% | 25% |

To improve the score we can return to the low-level data and examine the evaluation criteria. In this case we would find that a lower score was attributed to “Identifies and appropriately formulates the problem” than “Formulates appropriate computing requirements” so we may elect to focus our efforts on how to better teach students how to identify and formulate problems.

Data is collected in the fall and spring semesters, analyzed at the end of the spring semester, discussed at the beginning of the fall semester, and recommendations implemented that fall or spring if possible. The assessment cycle schedule is shown below.

A proposed programmatic change 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 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. A significantly modified assessment plan is to be forwarded to the Dean’s office, the College of Engineering Assessment Committee, the Faculty Senate Academic Assessment Committee, and the Office of Academic Affairs.

Program Educational Objectives

The CS program has also established Program Educational Objectives. Educational objectives are items that students should be able to accomplish within 5 years of graduation. ABET does not require assessment of the objectives, but we have a process in which they are examined through a survey of graduates and in a yearly meeting with constituents.

Appendix A: Faculty Review of Student Artifacts

**Measure Description:**

The student artifacts selected for assessment will vary depending upon the course and instructor but include assignments, exam questions, presentations, papers, design documents, requirements documents, and software.

Rubrics for each outcome are shown below.

**Outcome 1:** **Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.**

**Artifacts selected from CSCE A401 (Software Engineering) and CSCE A470 (Capstone)**

| **Evaluation Criteria** | **Poor** | **Developing** | **Satisfactory** | **Excellent** |
| --- | --- | --- | --- | --- |
| 1. Analyzes problem and formulates requirements for the problem | No attempt or fails to analyze accurately | Analyzes but key details are missing or confused | Most details analyzed and key relationships identified | Clearly analyzes the challenge and embedded issues |
| 1. Identifies solution by applying principles of computing | Incorrect application of computing principles or fails to identify solutions | Limited identification of solutions using computing principles | Reasonable identification of solutions using computing principles | In-depth and comprehensive utilization of computing principles, identification of solution well beyond expectations |

**Outcome 2:** **Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.**

**Artifacts selected from CSCE A401 (Software Engineering), CSCE A470 (Capstone)**

| **Evaluation Criteria** | **Poor** | **Developing** | **Satisfactory** | **Excellent** |
| --- | --- | --- | --- | --- |
| 1. Produces a design strategy, including tasks and subtasks, timelines, and evaluation of progress | Does not produce a design strategy, or the design strategy is especially poor | Limited attempts to form a design strategy | Produces a reasonable design strategy appropriate to the project | Produces an exceptional design strategy which exceeds expectations |
| 1. Creates a final product for evaluation | Does not create a final product, or the final product is especially poor | Makes a start on a final product but is unable to meet final specifications | Creates a satisfactory final product which meets defined specifications | Creates an exceptional final product which exceeds expectations |
| 1. Evaluates computing-based solution | Limited or no evaluation | Basic evaluation but has gaps | Satisfactory evaluation of solution, some utilization of computing principles (e.g. Big-O analysis, testing methodologies) | Exceptional and comprehensive evaluation of solution with strong tie to computing principles |

**Outcome 3:** **Communicate effectively in a variety of professional contexts, including technical and non-technical audiences for business, end-user, client, and computing contexts.**

**Artifacts selected from CSCE A401 (Software Engineering) and CSCE A470 (Capstone)**

| **Evaluation Criteria** | **Poor** | **Developing** | **Satisfactory** | **Excellent** |
| --- | --- | --- | --- | --- |
| 1. Effectively organizes and structures a presentation or document | No logical structure | Some structure but erratic jumps in topic | Most information presented logically | All information presented logically |
| 1. Provides appropriate content to demonstrate detailed knowledge of subject area | No grasp of topic, cannot answer questions or extremely limited content | Only rudimentary knowledge demonstrated | At ease with content and provides some detail | Full command of subject matter |
| 1. Effectively communicates details appropriate to the audience, including questions | Is unable to effectively communicate | Only able to answer/explain in a limited manner; limited detail | Provides sufficient detail to describe/answer questions | Communicates details exceptionally well |
| 1. Provides effective and appropriate visual aids and graphics | None | Weak support of the material, text or diagrams hard to see or understand | Mostly supports the material, most text and diagrams understandable | Text and diagrams strongly reinforce the presentation |
| 1. Writes using proper spelling and grammar | Significant errors | Several errors | Minor errors | Negligible errors |
| 1. Delivers oral presentation effectively | Significant delivery problems, little to no audience contact; much too long or much too short | Several mispronunciation, occasional audience contact; too long or too short | Clear voice, steady rate, some audience contact; slightly too long or too short | Clear voice, steady rate, strong audience contact, enthusiastic, confident; on time |

**Outcome 4:** **Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.**

**Artifacts selected from CSCE A465 (Computer Security)**

| **Evaluation Criteria** | **Poor** | **Developing** | **Satisfactory** | **Excellent** |
| --- | --- | --- | --- | --- |
| 1. Demonstrates the applicability of legal principles to the computing profession and impact on society, global, environmental, and economic contexts | Does not connect legal principles to computing practice and society and other contexts | Limited application of legal principles to computing practice and society and other contexts | Satisfactory application of legal principles to computing practice and society and other contexts | Strongly connects legal principles to computing practice and society and other contexts |
| 1. Demonstrates the applicability of ethical principles to the computing profession and impact on society, global, environmental, and economic contexts | Does not connect ethical principles to computing practice and society and other contexts | Limited application of ethical principles to computing practice and society and other contexts | Satisfactory application of ethical principles to computing practice and society and other contexts | Strongly connects ethical principles to computing practice and society and other contexts |

**Outcome 5:** **Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.**

**Artifacts selected from CSCE A401 (Software Engineering) and CSCE A470 (Capstone)**

| **Evaluation Criteria** | **Poor** | **Developing** | **Satisfactory** | **Excellent** |
| --- | --- | --- | --- | --- |
| 1. Understands and fulfills roles and responsibilities | Does not fulfill team role duties | Fulfills some, but not all, team role duties | Fulfills team role duties | Exceeds expectations with respect to team role duties |
| 1. Listens and works with others | Does not consider other team members’ ideas or concerns | Sometimes considers other team members’ ideas or concerns | Often addresses other team members’ ideas or concerns | Is exceptionally adept at addressing other team members’ ideas or concerns |
| 1. Communicates effectively with the group | Does not communicate to other members regarding the project progress | Provides terse outline of status of the project and relevant updates | Provides updates on a regular basis | Works exceptionally well to provide documentation of progress |

**Outcome 6:** **Apply computer science theory and software development fundamentals to produce computing-based solutions.**

**Artifacts selected from CSCE A351 (Automata & Algorithms) and ETS MFT**

| **Evaluation Criteria** | **Poor** | **Developing** | **Satisfactory** | **Excellent** |
| --- | --- | --- | --- | --- |
| 1. Worked through a theoretical problem on computing | Did not attempt the problem | Attempted the problem with the correct starting point but the work and final answer was incorrect | Attempted the problem with the correct approach but the final answer was incorrect | Answered the problem correctly with the correct approach |
| 1. Demonstrates expertise with algorithms and data structures | Algorithms or data structures employed incompatible with the problem | Inappropriate algorithms or data structures resulting in inefficiency or scalability problems | Mostly appropriate algorithms and data structures with some analysis for efficiency and scalability | Appropriate algorithms and data structures with analysis for efficiency and scalability |
| Performance on ETS Major Field Test in Computer Science (Theory, Software Fundamentals, Systems Fundamentals) | Lower 25th percentile | 25th to 50th percentile | 50th to 75th percentile | 75th to 100th percentile |

**Factors that affect the collected data:**

* Time and energy requirements. Significant work is required to collect and analyze the data.
* Bias associated with the grading philosophy of a single faculty. This is somewhat mitigated by use of the rubrics and by aggregation across several courses, student artifacts, and instructors.

**Tabulating, Interpretation, and Reporting Results**

A score of 1-4 is assigned to evaluations corresponding to Poor, Developing, Satisfactory, or Excellent. We will average data across each outcome and will drill down to averages and/or histograms of sub-outcomes as needed. An average score of “Poor” or “Developing” warrants remedial action while continuous improvement is possible for higher scores. The results will be discussed with the program faculty during the assessment meeting.

Appendix B: ETS Major Field Test In Computer Science

**Measure 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 test is administered during the CSCE A470 course. The test results feed into Outcome 6 and also can be analyzed for trends on their own.

**Factors that affect the collected data:**

* Student motivation. We do not 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. Each exam costs $30. If funding is unavailable we will be unable to collect data.
* Number of graduating students. A small number of graduating students taking the exam undermines the statistical validity of the results.
* While the data provides a useful snapshot as to our student performance compared to other institutions, the granularity of the results makes it difficult to effect change. For example, if our score is low in the area of Systems, then it is difficult to determine if the deficiency lies in Operating Systems, Architecture, Networking, etc.

**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, Computer Organization/Architecture/Operating Systems, and Algorithms/Theory/Computation/Math. Averages and percentiles are provided for all universities nationwide. The percentile is translated into a score using the rubrics described in Appendix A.

**Tabulating and Reporting Results:**

The test results will be received once a year. Based on the percentile for each student we will make a tabulation on the rubrics for Outcome 6.

We will also chart the department’s performance yearly and compare the performance to that of other departments nationwide.

Appendix C: Student Exit Survey

**Measure 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 in the CSCE A470 course. The students anonymously complete the surveys online.

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.

* Student knowledge. Students that enroll in CSCE 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:**

We must consider sample size and the indirect nature of the survey. Student feedback should be considered by the program faculty.

**Tabulating and Reporting Results:**

The survey is administered by the assessment coordinator. The assessment coordinator receives the results and tabulates them for use in outcomes review. The results are charted and tracked separately from the rubric evaluation of student artifacts.

**CS Exit Survey 18-19**

**Start of Block: Default Question Block**

Q1 All programs at UAA are required to implement an outcomes-based assessment program.  As a part of the assessment 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 community.

Q2 Do you intend to pursue graduate study?

* Yes
* No

Q3 Have you received any job offers, and if so, how many?

* No, because I did not apply for any jobs
* No, not yet
* 1
* 2
* 3
* 4
* 5+
* N/A, I am planning to go to graduate school

Q23 Have you accepted a permanent position, and if so, where?

* No, still looking
* I am going to graduate school (enter name if known) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* I have accepted a job offer (enter company name) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q24 If you have received a job offer and are willing to share the information with us, what is your starting salary?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q4 Primary area of Computer Science that you hope to work in:

* Software Development
* Networking or System Administration
* Research
* Management
* Technical Support
* Database Systems
* Software Testing
* Systems Analyst
* Not working in Computer Science
* Other

Q5 Please explain what other area of Computer Science you hope to work in.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q6 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 to each outcome.    
1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

|  | Poor | Fair | Good | Excellent | Outstanding | No Opinion |
| --- | --- | --- | --- | --- | --- | --- |
| What is your proficiency now? |  |  |  |  |  |  |
| How well did we do teaching this? |  |  |  |  |  |  |

Q7   2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.

|  | Poor | Fair | Good | Excellent | Outstanding | No Opinion |
| --- | --- | --- | --- | --- | --- | --- |
| What is your proficiency now? |  |  |  |  |  |  |
| How well did we do teaching this? |  |  |  |  |  |  |

Q8 3)  Communicate effectively in a variety of professional contexts, including technical and non-technical audiences for business, end-user, client, and computing contexts.

|  | Poor | Fair | Good | Excellent | Outstanding | No Opinion |
| --- | --- | --- | --- | --- | --- | --- |
| What is your proficiency now? |  |  |  |  |  |  |
| How well did we do teaching this? |  |  |  |  |  |  |

Q9 4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

|  | Poor | Fair | Good | Excellent | Outstanding | No Opinion |
| --- | --- | --- | --- | --- | --- | --- |
| What is your proficiency now? |  |  |  |  |  |  |
| How well did we do teaching this? |  |  |  |  |  |  |

Q10 5) Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.

|  | Poor | Fair | Good | Excellent | Outstanding | No Opinion |
| --- | --- | --- | --- | --- | --- | --- |
| What is your proficiency now? |  |  |  |  |  |  |
| How well did we do teaching this? |  |  |  |  |  |  |

Q11 6)  Apply computer science theory and software development fundamentals to produce computing-based solutions.

|  | Poor | Fair | Good | Excellent | Outstanding | No Opinion |
| --- | --- | --- | --- | --- | --- | --- |
| What is your proficiency now? |  |  |  |  |  |  |
| How well did we do teaching this? |  |  |  |  |  |  |

Q12 Please indicate your satisfaction with each of the following aspects of your experience at UAA.

|  | Poor | Fair | Good | Excellent | Outstanding | No Opinion |
| --- | --- | --- | --- | --- | --- | --- |
| Quality of the Advising |  |  |  |  |  |  |
| Quality of the Instruction |  |  |  |  |  |  |
| Quality of Computer Laboratories |  |  |  |  |  |  |
| Quality of Physical Facilities (other than computing labs) |  |  |  |  |  |  |

Q13 Please add any optional explanation for the previous question (Quality of the Advising, Instruction, Computer Labs, or Physical Facilities).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q14 Please list up to three major strengths of your undergraduate computer science education or other UAA experiences.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q15 Please list up to three areas for improvement in our undergraduate computer science program or other aspects of UAA.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q16 With respect to the previous question, do you have any suggestions on how UAA could address these improvements?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q17 Would you recommend a UAA computer science education to a friend or relative?

* Yes
* Maybe
* No

**End of Block: Default Question Block**