# Program Outcomes Assessment

**MA/MS in Computer Science**

**Created on: 10/06/2010 07:17:00 AM CDT**
**Last Modified: 03/25/2015 07:28:36 AM CDT**

# Table of Contents

# General Information (Program Outcomes Assessment)

# Standing Requirements

## Mission Statement

The graduate computer science program at Indiana State University is a master'sl computer science program that provides equal emphasis on providing students with a theoretical foundation for the subject and ensuring students have in depth programming experience. The program aims to ensure students are well-prepared either for (a) employment in software and technology positions, or (b) further studies in a PhD program in computer science. The program provides more in depth training in algorithm analysis and development than the undergraduate program. The graduate program also emphasizes more independent work, including projects.

## Outcomes Library

### MA/MS in Computer Science Outcome Set - Effective 2012-2013

#### Objective 1 - Reinforce Core CS Skills

One objective of the master's program is to reinforce the core CS skills that students become proficient at as undergraduates. If there are any deficiencies in students' core CS skills, these should be removed by the time a student graduates from the master's program. See the Program Educational Objectives and Learning Outcomes for the undergraduate program for more details.

| Outcome | Mapping |
|---|---|
| 1A - Reinforce Core CS Skills | No Mapping |

#### Objective 2 - Algorithms and Analysis

The second main objective is that students have more specialized knowledge of algorithms and analysis than that expected of undergraduate students.

| Outcome | Mapping |
|---|---|
| 2A - Algorithms<br>Students are exposed to a wider variety of algorithm techniques than in the undergraduate program, including a more detailed look at linear programming, randomized algorithms, and the computational complexity of intractable problems. Students gain experience programming a wider variety of algorithms, and should be proficient at analyzing new problems in light of the techniques they have learned. | No Mapping |
| 2B - Large Software Projects<br>Students gain experience working on large software projects, using libraries and techniques that are more on the cutting edge of present research. Students should be proficient at working on large software projects. | No Mapping |
| 2C - Independent Research<br>Students should be proficient at independently researching the best algorithms for a given problem, understanding the state of the art algorithms, and programming or using existing tools to solve the problem. | No Mapping |

### Objective 3 - Professional, Interpersonal, and Writing Skills

Another objective of the program is that the students develop professional, interpersonal, and writing skills.

| Outcome | Mapping |
|---|---|
| 3A - Working in a Group<br>Students should have had some experience working on a programming project in a group setting. | No Mapping |
| 3B - Presentation Skills<br>Students should have given presentations in some of their classes, and should be aware of and have mastered basic presentation skills. | No Mapping |
| 3C - Writing Skills<br>Students should have experience writing a technical research paper. Students should be knowledgeable in the standard formatting and style for such papers. | No Mapping |

## Replaced - MA/MS in Computer Science Outcome Set

### 1. Knowledge of advanced software development techniques

Demonstrate knowledge of advanced software development techniques.

| Outcome | Mapping |
|---|---|
| 1A. Perform object oriented software development<br>Students should be able to perform object oriented software development, component design and architecture design. | No Mapping |
| 1B. Describe and utilize design patterns.<br>Students should be able to describe and utilize design patterns. | No Mapping |
| 1C. Describe and utilize XML.<br>Students should be able to describe and utilize XML. | No Mapping |

### 2. knowledge of advanced concurrent processing techniques.

Demonstrate knowledge of advanced concurrent processing techniques.

| Outcome | Mapping |
|---|---|
| 2A. Describe and utilize shared variables.<br>Students should be able to describe and utilize shared variables. | No Mapping |
| 2B. Describe and utilize message passing.<br>Students should be able to describe and utilize message passing. | No Mapping |
| 2C. Describe and utilize concurrent programming languages<br>Students should be able to describe and utilize concurrent programming languages. | No Mapping |

### 3. Knowledge of advanced topics in the theory of computing.

Demonstrate knowledge of advanced topics in the theory of computing.

| Outcome | Mapping |
|---|---|
| 3A. Algorithm analysis and design.<br>Students should be able to describe and utilize advanced | No Mapping |

| | |
|---|---|
| methods of algorithm analysis and design. | |
| 3B. Formal languages, logic, automata and computability<br>Students should be able to describe and utilize advanced topics in formal languages, logic, automata and computability | No Mapping |
| 3C. Describe & utilize advanced topics in discrete structure<br>Students should be able to describe and utilize advanced topics in discrete structures. | No Mapping |

## Curriculum Map

### Active Curriculum Maps

**MA/MS in Computer Science Cirriculum Map - Effective 2012-2013** (See appendix)
**Alignment Set:** MA/MS in Computer Science Outcome Set - Effective 2012-2013

**Created:** 03/24/2015 2:39:57 pm CDT
**Last Modified:** 03/24/2015 2:54:36 pm CDT

**Replaced - MA/MS in Computer Science Cirriculum Map** (See appendix)
**Alignment Set:** Replaced - MA/MS in Computer Science Outcome Set

**Created:** 02/28/2012 9:14:55 am CDT
**Last Modified:** 03/24/2015 2:39:34 pm CDT

## Communication of Outcomes

The department regularly assesses the success of the program in meeting its mission in general and in meeting program objectives in particular. These assessments are communicated with students and other stakeholders via the department website.

# Archive (This area is to be used for archiving pre-TaskStream assessment data and for current documents.)

## Archive

# 2011-2012 Assessment Cycle

## Assessment Plan

### Outcomes and Measures

### Replaced - MA/MS in Computer Science Outcome Set

**2. knowledge of advanced concurrent processing techniques.**

Demonstrate knowledge of advanced concurrent processing techniques.

**2A. Describe and utilize shared variables.**

Students should be able to describe and utilize shared variables.

**Measure:** Exam
Direct - Exam

**Details/Description:** Students must be able to correctly interpret the output of programs such as {\tt ps}, and determine memory use, CPU use, and status of processes.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that start and control multiple processes, using {\tt fork} and {\tt exec*} system calls.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test

**Details/Description:** Students must be able to write multi-process programs that use shared memory.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that do interprocess communication via message passing, using, for example, POSIX message queues.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that create and effectively use threads to solve non-trivial problems.
The effectiveness of such a program is measured by determining whether or not the use of threads has reduced the running time.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test

**Details/Description:** Students must be able take a linear single threaded algorithm and effectively rewrite it as a multiple-threaded algorithms. Example algorithms where this is possible would include graph algorithms such as maximum clique, maximum independent set, chromatic number, hamiltonian cycle, domination number, girth, diameter, etc.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**2B. Describe and utilize message passing.**

Students should be able to describe and utilize message passing.

**Measure:** Exam
Direct - Exam

**Details/Description:** Students must be able to correctly interpret the output of programs such as {\tt ps}, and determine memory use, CPU use, and status of processes.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that do elementary interprocess communication via pipes and signals.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that do interprocess communication via message passing, using, for example, POSIX message queues

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that create and effectively use threads to solve non-trivial problems.
The effectiveness of such a program is measured by determining whether or not the use of threads has reduced the running time.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able take a linear single threaded algorithm and effectively rewrite it as a multiple-threaded algorithms. Example algorithms where this is possible would include graph algorithms such as maximum clique, maximum independent set, chromatic number, hamiltonian cycle, domination number, girth, diameter, etc.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**2C. Describe and utilize concurrent programming languages**

Students should be able to describe and utilize concurrent programming languages.

**Measure:** Exam
Direct - Exam

**Details/Description:** Students must be able to correctly interpret the output of programs such as {\tt ps}, and determine memory use, CPU use, and status of processes.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that start and control multiple processes, using {\tt fork} and {\tt exec*} system calls.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write multi-process programs that use shared memory.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that create and effectively use threads to solve non-trivial problems. The effectiveness of such a program is measured by determining whether or not the use of threads has reduced the running time.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able take a linear single threaded algorithm and effectively rewrite it as a multiple-threaded algorithms. Example algorithms where this is possible would include graph algorithms such as maximum clique, maximum independent set, chromatic number, hamiltonian cycle, domination number, girth, diameter, etc.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

## Assessment Findings

### Finding per Measure

### Replaced - MA/MS in Computer Science Outcome Set

**2. knowledge of advanced concurrent processing techniques.**

Demonstrate knowledge of advanced concurrent processing techniques.

**2A. Describe and utilize shared variables.**

Students should be able to describe and utilize shared variables.

**Measure:** Exam
Direct - Exam

**Details/Description:** Students must be able to correctly interpret the output of programs such as {\tt ps}, and determine memory use, CPU use, and status of processes.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Exam

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that start and control multiple processes, using {\tt fork} and {\tt exec*} system calls.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test

**Details/Description:** Students must be able to write multi-process programs that use shared memory.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that do interprocess communication via message passing, using, for example, POSIX message queues.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that create and effectively use threads to solve non-trivial problems. The effectiveness of such a program is measured by determining whether or not the use of threads has reduced the running time.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test

**Details/Description:** Students must be able take a linear single threaded algorithm and effectively rewrite it as a multiple-threaded algorithms. Example algorithms where this is possible would include graph algorithms such as maximum clique, maximum independent set, chromatic number, hamiltonian cycle, domination number, girth, diameter, etc.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**2B. Describe and utilize message passing.**

Students should be able to describe and utilize message passing.

**Measure:** Exam
Direct - Exam

**Details/Description:** Students must be able to correctly interpret the output of programs such as {\tt ps}, and determine memory use, CPU use, and status of processes.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Exam

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that do elementary interprocess communication via pipes and signals.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that do interprocess communication via message passing, using, for example,

POSIX message queues

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that create and effectively use threads to solve non-trivial problems.
The effectiveness of such a program is measured by determining whether or not the use of threads has reduced the running time.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able take a linear single threaded algorithm and effectively rewrite it as a multiple-threaded algorithms. Example algorithms where this is possible would include graph algorithms such as maximum clique, maximum independent set, chromatic number, hamiltonian cycle, domination number, girth, diameter, etc.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**2C. Describe and utilize concurrent programming languages**

Students should be able to describe and utilize concurrent programming languages.

**Measure:** Exam
Direct - Exam

**Details/Description:** Students must be able to correctly interpret the output of programs such as {\tt ps}, and determine memory use, CPU use, and status of processes.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Exam

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that start and control multiple processes, using {\tt fork} and {\tt exec*} system calls.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write multi-process programs that use shared memory.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able to write programs that create and effectively use threads to solve non-trivial problems. The effectiveness of such a program is measured by determining whether or not the use of threads has reduced the running time.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

**Measure:** Program Test
Direct - Student Artifact

**Details/Description:** Students must be able take a linear single threaded algorithm and effectively rewrite it as a multiple-threaded algorithms. Example algorithms where this is possible would include graph algorithms such as maximum clique, maximum independent set,

chromatic number, hamiltonian cycle, domination number, girth, diameter, etc.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Program Test

*No Findings Added*

## Overall Recommendations

*No text specified*

## Overall Reflection

*No text specified*

# 2012-2013 Assessment Cycle

## Assessment Plan

### Outcomes and Measures

## Assessment Findings

### Finding per Measure

### Overall Recommendations

*No text specified*

### Overall Reflection

*No text specified*

## Action Plan

## Status Report

# 2013-2014 Assessment Cycle

## Assessment Plan

### Outcomes and Measures

### MA/MS in Computer Science Outcome Set - Effective 2012-2013

#### Objective 1 - Reinforce Core CS Skills

One objective of the master's program is to reinforce the core CS skills that students become proficient at as undergraduates. If there are any deficiencies in students' core CS skills, these should be removed by the time a student graduates from the master's program. See the Program Educational Objectives and Learning Outcomes for the undergraduate program for more details.

**1A - Reinforce Core CS Skills**

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

#### Objective 2 - Algorithms and Analysis

The second main objective is that students have more specialized knowledge of algorithms and analysis than that expected of undergraduate students.

**2A - Algorithms**

Students are exposed to a wider variety of algorithm techniques than in the undergraduate program, including a more detailed look at linear programming, randomized algorithms, and the computational complexity of intractable problems. Students gain experience programming a wider variety of algorithms, and should be proficient at analyzing new problems in light of the techniques they have learned.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)

Exit Survey (Word Document (Open XML)) (See appendix)

**2B - Large Software Projects**

Students gain experience working on large software projects, using libraries and techniques that are more on the cutting edge of present research. Students should be proficient at working on large software projects.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)

Exit Survey (Word Document (Open XML)) (See appendix)

**2C - Independent Research**

Students should be proficient at independently researching the best algorithms for a given problem, understanding the state of the art algorithms, and programming or using existing tools to solve the problem.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

## Objective 3 - Professional, Interpersonal, and Writing Skills

Another objective of the program is that the students develop professional, interpersonal, and writing skills.

### 3A - Working in a Group

Students should have had some experience working on a programming project in a group setting.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

### 3B - Presentation Skills

Students should have given presentations in some of their classes, and should be aware of and have mastered basic presentation skills.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

- Alumni Survey (Word Document (Open XML)) (See appendix)
- Exit Survey (Word Document (Open XML)) (See appendix)

**3C - Writing Skills**

Students should have experience writing a technical research paper. Students should be knowledgeable in the standard formatting and style for such papers.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

- Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

- Alumni Survey (Word Document (Open XML)) (See appendix)
- Exit Survey (Word Document (Open XML)) (See appendix)

## Assessment Findings

### Finding per Measure

### MA/MS in Computer Science Outcome Set - Effective 2012-2013

**Objective 1 - Reinforce Core CS Skills**

One objective of the master's program is to reinforce the core CS skills that students become proficient at as undergraduates. If there are any deficiencies in students' core CS skills, these should be removed by the time a student graduates from the master's program. See the Program Educational Objectives and Learning Outcomes for the undergraduate program for more details.

**1A - Reinforce Core CS Skills**

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

## Objective 2 - Algorithms and Analysis

The second main objective is that students have more specialized knowledge of algorithms and analysis than that expected of undergraduate students.

### 2A - Algorithms

Students are exposed to a wider variety of algorithm techniques than in the undergraduate program, including a more detailed look at linear programming, randomized algorithms, and the computational complexity of intractable problems. Students gain experience programming a wider variety of algorithms, and should be proficient at analyzing new problems in light of the techniques they have learned.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

### 2B - Large Software Projects

Students gain experience working on large software projects, using libraries and techniques that are more on the cutting edge of present research. Students

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

should be proficient at working on large software projects.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

**2C - Independent Research**

Students should be proficient at independently researching the best algorithms for a given problem, understanding the state of the art algorithms, and programming or using existing tools to solve the problem.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

## Objective 3 - Professional, Interpersonal, and Writing Skills

Another objective of the program is that the students develop professional, interpersonal, and writing skills.

### 3A - Working in a Group

Students should have had some experience working on a programming project in a group setting.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

### 3B - Presentation Skills

Students should have given presentations in

**Measure:** Research Project - CS 695
Direct - Student Artifact

some of their classes, and should be aware of and have mastered basic presentation skills.

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

**3C - Writing Skills**

Students should have experience writing a technical research paper. Students should be knowledgeable in the standard formatting and style for such papers.

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit

survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

- Alumni Survey (Word Document (Open XML)) (See appendix)
- Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

## Overall Recommendations

*No text specified*

## Overall Reflection

*No text specified*

## Action Plan

## Status Report

# 2014-2015 Assessment Cycle

- **Assessment Plan**
- **Assessment Findings**
- **Action Plan**
- **Status Report**

# 2015-2016 Assessment Cycle

## Assessment Plan

### Outcomes and Measures

### MA/MS in Computer Science Outcome Set - Effective 2012-2013

#### Objective 1 - Reinforce Core CS Skills

One objective of the master's program is to reinforce the core CS skills that students become proficient at as undergraduates. If there are any deficiencies in students' core CS skills, these should be removed by the time a student graduates from the master's program. See the Program Educational Objectives and Learning Outcomes for the undergraduate program for more details.

**1A - Reinforce Core CS Skills**

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** Graduate students' mastery of core CS skills is measured by their performance in their research project in the course CS 695 Computer Science Research, and through periodic testing of students using the same testing mechanisms as the used to assess the undergraduate students' mastery of core CS skills. It is expected that all students graduating from the master's program have full mastery of all core CS skills by the time they finish their master's degree.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

#### Objective 2 - Algorithms and Analysis

The second main objective is that students have more specialized knowledge of algorithms and analysis than that expected of undergraduate students.

**2A - Algorithms**

Students are exposed to a wider variety of algorithm techniques than in the undergraduate program, including a more detailed look at linear programming, randomized algorithms, and the computational complexity of intractable problems. Students gain experience programming a wider variety of algorithms, and should be proficient at analyzing new problems in light of the techniques they have learned.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**2B - Large Software Projects**

Students gain experience working on large software projects, using libraries and techniques that are more on the cutting edge of present research. Students should be proficient at working on large software projects.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**2C - Independent Research**

Students should be proficient at independently researching the best algorithms for a given problem, understanding the state of the art algorithms, and programming or using existing tools to solve the problem.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

## Objective 3 - Professional, Interpersonal, and Writing Skills

Another objective of the program is that the students develop professional, interpersonal, and writing skills.

### 3A - Working in a Group

Students should have had some experience working on a programming project in a group setting.

#### Measure: Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

#### Measure: Surveys

Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

### 3B - Presentation Skills

Students should have given presentations in some of their classes, and should be aware of and have mastered basic presentation skills.

#### Measure: Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

#### Measure: Surveys

Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**3C - Writing Skills**

Students should have experience writing a technical research paper. Students should be knowledgeable in the standard formatting and style for such papers.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

## Assessment Findings

### Finding per Measure

### MA/MS in Computer Science Outcome Set - Effective 2012-2013

### Objective 1 - Reinforce Core CS Skills

One objective of the master's program is to reinforce the core CS skills that students become proficient at as undergraduates. If there are any deficiencies in students' core CS skills, these should be removed by the time a student graduates from the master's program. See the Program Educational Objectives and Learning Outcomes for the undergraduate program for more details.

**1A - Reinforce Core CS Skills**

**Measure:** Research Project - CS 695
Direct - Student Artifact

**Details/Description:** Graduate students' mastery of core CS skills is measured by their performance in their research project in the course CS 695 Computer Science Research, and through periodic testing of students using the same testing mechanisms as the used to assess the undergraduate students' mastery of core CS skills. It is expected that all students graduating from the master's program have full mastery of all core CS skills by the time they finish their master's degree.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Findings** for Research Project - CS 695

*No Findings Added*

## Objective 2 - Algorithms and Analysis

The second main objective is that students have more specialized knowledge of algorithms and analysis than that expected of undergraduate students.

### 2A - Algorithms

Students are exposed to a wider variety of algorithm techniques than in the undergraduate program, including a more detailed look at linear programming, randomized algorithms, and the computational complexity of intractable problems. Students gain experience programming a wider variety of algorithms, and should be proficient at analyzing new problems in light of the techniques they have learned.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

### 2B - Large Software Projects

Students gain experience working on large software projects, using libraries and techniques that are more on the cutting edge of present research. Students should be proficient at working on large software projects.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

**2C - Independent Research**

Students should be proficient at independently researching the best algorithms for a given problem, understanding the state of the art algorithms, and programming or using existing tools to solve the problem.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

## Objective 3 - Professional, Interpersonal, and Writing Skills

Another objective of the program is that the students develop professional, interpersonal, and writing skills.

### 3A - Working in a Group

Students should have had some experience working on a programming project in a group setting.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

### 3B - Presentation Skills

Students should have given presentations in some of their classes, and should be aware of and have mastered basic presentation skills.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

**3C - Writing Skills**

Students should have experience writing a technical research paper. Students should be knowledgeable in the standard formatting and style for such papers.

**Measure:** Research Project - CS 695

**Details/Description:** The outcomes of Objective 2 are measured by evaluating a student's research project for the course CS 695 Computer Science Research. A sample evaluation form is given in the appendices.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Research Project CS 695 Evaluation Rubric (Word Document (Open XML)) (See appendix)

**Findings** for Research Project - CS 695

*No Findings Added*

**Measure:** Surveys
Indirect - Survey

**Details/Description:** The student's own assessment of their mastery is obtained in an exit survey.

**Target:**

**Implementation Plan (timeline):**

**Responsible Individual(s):**

**Supporting Attachments:**

Alumni Survey (Word Document (Open XML)) (See appendix)
Exit Survey (Word Document (Open XML)) (See appendix)

**Findings** for Surveys

*No Findings Added*

### Overall Recommendations

*No text specified*

### Overall Reflection

*No text specified*

## Action Plan

## Status Report

# 2016-2017 Assessment Cycle

- **Assessment Plan**
- **Assessment Findings**

# 2017-2018 Assessment Cycle

- **Assessment Plan**
- **Assessment Findings**

# 2018-2019 Assessment Cycle

- **Assessment Plan**
- **Assessment Findings**

# 2019-2020 Assessment Cycle

- **Assessment Plan**
- **Assessment Findings**

# Appendix

A. **MA/MS in Computer Science Cirriculum Map - Effective 2012-2013** (Curriculum Map)
B. **Replaced - MA/MS in Computer Science Cirriculum Map** (Curriculum Map)
C. **Alumni Survey** (Word Document (Open XML))
D. **Alumni Survey** (Word Document (Open XML))
E. **Alumni Survey** (Word Document (Open XML))
F. **Alumni Survey** (Word Document (Open XML))
G. **Alumni Survey** (Word Document (Open XML))
H. **Exit Survey** (Word Document (Open XML))
I. **Exit Survey** (Word Document (Open XML))
J. **Exit Survey** (Word Document (Open XML))
K. **Exit Survey** (Word Document (Open XML))
L. **Exit Survey** (Word Document (Open XML))
M. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
N. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
O. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
P. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
Q. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
R. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
S. **Alumni Survey** (Word Document (Open XML))
T. **Alumni Survey** (Word Document (Open XML))
U. **Alumni Survey** (Word Document (Open XML))
V. **Alumni Survey** (Word Document (Open XML))
W. **Alumni Survey** (Word Document (Open XML))
X. **Alumni Survey** (Word Document (Open XML))
Y. **Alumni Survey** (Word Document (Open XML))

Z. **Alumni Survey** (Word Document (Open XML))
AA. **Alumni Survey** (Word Document (Open XML))
AB. **Alumni Survey** (Word Document (Open XML))
AC. **Alumni Survey** (Word Document (Open XML))
AD. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AE. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AF. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AG. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AH. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AI. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AJ. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AK. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AL. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AM. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AN. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AO. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
AP. **Exit Survey** (Word Document (Open XML))
AQ. **Exit Survey** (Word Document (Open XML))
AR. **Exit Survey** (Word Document (Open XML))
AS. **Exit Survey** (Word Document (Open XML))
AT. **Exit Survey** (Word Document (Open XML))
AU. **Exit Survey** (Word Document (Open XML))
AV. **Exit Survey** (Word Document (Open XML))
AW. **Exit Survey** (Word Document (Open XML))
AX. **Exit Survey** (Word Document (Open XML))
AY. **Exit Survey** (Word Document (Open XML))
AZ. **Exit Survey** (Word Document (Open XML))
BA. **Exit Survey** (Word Document (Open XML))
BB. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
BC. **Alumni Survey** (Word Document (Open XML))
BD. **Alumni Survey** (Word Document (Open XML))
BE. **Alumni Survey** (Word Document (Open XML))
BF. **Alumni Survey** (Word Document (Open XML))
BG. **Alumni Survey** (Word Document (Open XML))
BH. **Alumni Survey** (Word Document (Open XML))
BI. **Exit Survey** (Word Document (Open XML))
BJ. **Exit Survey** (Word Document (Open XML))
BK. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
BL. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
BM. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
BN. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
BO. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
BP. **Exit Survey** (Word Document (Open XML))
BQ. **Exit Survey** (Word Document (Open XML))
BR. **Exit Survey** (Word Document (Open XML))
BS. **Research Project CS 695 Evaluation Rubric** (Word Document (Open XML))
BT. **Exit Survey** (Word Document (Open XML))
BU. **Alumni Survey** (Word Document (Open XML))

## Exit survey

The following is a survey that will be given to students as they graduate.

The following questions are an anonymous survey from the Department of Mathematics and Computer Science at Indiana State University. The department values the opinions and experiences of our alumni. We very much appreciate you taking time to answer these questions. We will use this information to improve the education of current and future students.

The survey will take between 5-15 minutes, depending on how many of the text-entry questions you answer. If you are limited on time, please just answer as many of the questions as you have time for. We would rather you answer a few questions than not complete the survey at all.

If you feel uncomfortable about answering any of the questions, please just leave that question blank/un-answered.

1. You identify your gender as: Male, Female, Trans/Other
2. Please select which degrees you earned at Indiana State: Associate, Bachelor, Master, PhD
3. Please select what your major was at Indiana State. If you had more than one major, then select each.
4. Please select any minors or certificates that you had at Indiana State. If you had more than one minor or certificate, then select each.
5. What are your plans right now? Select all that apply.
    - Seeking employment related to programming/CS/IT.
    - Found employment related to programming/CS/IT.
    - Seeking employment not related to programming/CS/IT.
    - Found employment not related to programming/CS/IT.
    - Not seeking employment.
    - Going to grad school in field related to CS.
    - Going to grad school in other field.
    - Not sure.
6. For each of the following, rate how well you feel you have mastered the skill. Ratings are Full mastery, Mostly mastered, Somewhat mastered, Not mastered at all, NA.
    - Basic programming
    - Programming in multiple different paradigms
    - Data structures
    - Algorithms analysis, including mathematics needed (e.g., basic graph theory, set theory, number theory, probability)
    - Computer systems, including operating systems and architecture
    - Software design processes, including software engineering and use of a variety of programming languages and environments
    - Program design in databases

    - Web programming
    - Program design in networks
    - Program design in graphics
    - Program design in artificial intelligence
    - Linux/Unix administration
    - Algorithm design techniques
    - Understanding models of computation
    - Formal methods
    - Working/programming in groups
    - Independent research of problems and algorithms
    - Presentation skills
    - Writing skills
7. Are there any of the CS/Math courses at ISU that you wish had covered some material better than they did, so that you would have been better prepared for later courses or for your job or later studies? If so, select which course(s)
8. Please select any of the following that you think the department needs to do a better job at.
    - Advising, helping students select courses
    - Helping students find internships
    - Programming - having enough programming in the courses so students leave ISU as good programmers.
    - Programming - exposing students to a wide range of programming languages and environments.
    - Math/Theory/Algorithms - having enough theory/algorithms in the courses so that students leave ISU as good problem solvers with knowledge of algorithms and how to apply them.
    - Projects - experience with big software projects, either in courses or as independent study or research.
    - Other not listed here.
9. Please select any of the following that you think the department did a good job at while you were at ISU. [Same choices as last question.]
10. Are there any topics, skills, etc. that were not covered in your CS/Math classes at ISU that you think should be covered more? If so, please list the most important ones.
11. If you were starting your degree again, would you choose ISU again?
    Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree
12. If you were starting over, would you choose the same major(s) again?
    Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree
13. Would you recommend ISU to someone you know who wants to go to school for CS/Math?
    Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree
14. Is there anything else you would like to let us know?

## Alumni survey

The following survey is sent to alumni at least once every five years.

The following questions are an anonymous survey from the Department of Mathematics and Computer Science at Indiana State University. The department values the opinions and experiences of our alumni. We very much appreciate you taking time to answer these questions. We will use this information to improve the education of current and future students.

The survey will take between 5-15 minutes, depending on how many of the text-entry questions you answer. If you are limited on time, please just answer as many of the questions as you have time for. We would rather you answer a few questions than not complete the survey at all.

If you feel uncomfortable about answering any of the questions, please just leave that question blank/un-answered.

1. You identify your gender as: Male, Female, Trans/Other
2. Please select which degrees you earned at Indiana State: Associate, Bachelor, Master, PhD
3. What year did you finish your last degree at Indiana State?
4. Please select what your major was at Indiana State. If you had more than one major, then select each.
5. Please select any minors or certificates that you had at Indiana State. If you had more than one minor or certificate, then select each.
6. What was your status in the first year after graduating from ISU? Select all that apply.
    - Sought employment unsuccessfully
    - Did not seek employment or additional education (e.g., stay at home parent)
    - Attended graduate school at another school
    - Employed - Academic/Professor
    - Employed - Education/Teaching
    - Employed - Programming
    - Employed - Systems analyst
    - Employed - Database administrator
    - Employed - Web programmer
    - Employed - Other related to programming or information technology
    - Employed - Not related to programming or information technology
7. How long did you look for employment before getting your first job after graduating from ISU? < 3 months, 3 - 6 months, 6 - 9 months, 9 - 12 months
8. (If grad school) What school did you go to graduate school at?
9. (If grad school) What degree did you obtain at that school? Masters, PhD, Other, Did not earn a degree
10. What is your current employment status? Select all that apply. [Same choices as above.]
11. If you have worked, what are a few of the companies/organizations that you have

worked for?

12. If you are currently working, what is your current salary?
    $0 to $30,000, $30,000 to $50,000, $50,000 to $70,000, $70,000 to $100,000, $100,000 to $150,000, $150,000 to $300,000, more than $300,000
13. If you have remained in a job related to computer science, which of your courses at ISU were MOST beneficial and relevant to your work?
14. Are there any of the CS/Math courses at ISU that you wish had covered some material better than they did, so that you would have been better prepared for later courses or for your job or later studies? If so, select which course(s)
15. If you have remained in a job related to computer science, which of your courses at ISU were LEAST beneficial and relevant to your work?
16. Please select any of the following that you think the department needs to do a better job at.
    - Advising, helping students select courses
    - Helping students find internships
    - Programming - having enough programming in the courses so students leave ISU as good programmers.
    - Programming - exposing students to a wide range of programming languages and environments.
    - Math/Theory/Algorithms - having enough theory/algorithms in the courses so that students leave ISU as good problem solvers with knowledge of algorithms and how to apply them.
    - Projects - experience with big software projects, either in courses or as independent study or research.
    - Other not listed here.
17. Please select any of the following that you think the department did a good job at while you were at ISU. [Same choices as last question.]
18. Are there any topics, skills, etc. that were not covered in your CS/Math classes at ISU that you think should be covered more? If so, please list the most important ones.
19. If you were starting your degree again, would you choose ISU again?
    Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree
20. If you were starting over, would you choose the same major(s) again?
    Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree
21. Would you recommend ISU to someone you know who wants to go to school for CS/Math?
    Strongly Disagree, Disagree, Neither Agree nor Disagree, Agree, Strongly Agree
22. Is there anything else you would like to let us know?

## CS 695 Evaluation Form

The form is filled out by two different faculty members at the completion of the student's research project and presentation. The student is rated with regard to the project they have worked on. The faculty members will ask the student verbal questions as needed to determine the student's mastery of each area.

1. Student last name, first name:
2. Year student entered:
3. Year student will graduate:
4. Project title:
5. Rate student skills
    a. Basic programming:
        - No mastery: Project does not include a working algorithm.
        - Some mastery: Project includes a working algorithm of some sort. Code may be unorganized. Student may show lack of understanding of some parts of the code.
        - Full mastery: Project includes a working algorithm that is among the most efficient for the given problem. Code is clean and well-documented. Student shows complete understanding of the code.
    b. Data structures:
        - No mastery: Student cannot answer even basic questions about the data structures used to solve the problem they studied.
        - Some mastery: Student can answer some questions about the data structures used to solve the problem they have studied, but is unable to answer all questions. Some explanations may be simply from memory without showing full understanding.
        - Full mastery: Student can answer nearly all questions about data structures used to solve the problem, and gives very good explanations.
    c. Algorithms and analysis:
        - No mastery: Student cannot explain an algorithm to solve the problem.
        - Some mastery: Student can explain one algorithm to solve the problem, but does not have an understanding of the range of algorithms available to solve the problem.
        - Full mastery: Student explains the efficiency of algorithms for the problem very well, including multiple different algorithms to solve the problem, and tradeoffs between different algorithms.
    d. Ability to work on large software project:
        - No mastery: Student was not able to implement an algorithm on their own.
        - Some mastery: The algorithm implemented is among the more simple.
        - Full mastery: The algorithm(s) implemented by the student is (are) among the the best available, and coding the algorithm involved a large amount of programming.

e. Ability to research independently:
    - No mastery: Student could not work independently on the project.
    - Some mastery: Student worked independently at times, but needed significant guidance for the instructors.
    - Full mastery: Student worked mostly independently on their project, needing little guidance from the instructors.
f. Presentation skills:
    - No mastery: Presentation shows a lack of understanding, or a lack of ability to explain concepts to someone unfamiliar with the problem studied.
    - Some mastery: Presentation is clear at times and not clear at others.
    - Full mastery: Presentation is very clear, explaining intuition, giving examples, and going at a pace comfortable to someone unfamiliar with the problem the student studied.
g. Writing skills:
    - No mastery: Project paper is not well organized, and shows a clear lack of understanding on the part of the student.
    - Some mastery: Project paper gives good basic understanding of the problem studied, but fails at times to give detailed understanding of the problem.
    - Full mastery: Project paper is well-written and organized, explains the problem studied and algorithms well, including the tradeoffs between different algorithms to solve the problem.

6. Other comments