Please complete this page for each degree program, graduate and undergraduate.

Student Learning Outcomes Check Sheet

Due 24 Feb 2012

Degree Program Title: **BS Computer Science**  Degree Type: ______________
Banner/CIP Code: ______________

Responsible Program Coordinator/Chair completing this form: SAFFER

A. Program MISSION Statement: What body of knowledge and/or what skills and qualities will graduates from this program possess upon completion of the degree?

**Mission Statement:**
The role of the Department of Computer Science and Information Systems is to maintain and administer two undergraduate academic programs and one graduate academic program in computer science and information systems. The purpose of these academic programs is to educate students pursuing a career in the varied fields of computer science and information technology; to educate quality computer science teachers for service in the public schools and institutions of higher education; to prepare students for advanced graduate study; to support the technological educational and training needs of local industry; to provide basic computer science service courses for the University. Furthermore, it is the goal of this Department to create an environment in which students will develop their intellectual curiosity, analytical abilities, and computational and communication skills in preparation for independent and life-long learning. It is desired that this department become an important educational resource for Northeast Texas attracting students and research activities from industry and business.

B. Does this program have any culminating experience or capstone course that would capture the cumulative knowledge and accomplishments of graduates of your program? If so, please describe the process by which faculty participate in the design and evaluation of the course and its products/experiences.

In the CS and CIS programs, we have a 2 course senior sequence (CSCI 359 & CSCI 440). Students are required to design and implement a 2 semester project that captures the cumulative knowledge and accomplishments. The project can include computer programming, web page design, database and SQL subprojects, web database interface, and more.

In the attached pages, please provide the learning outcomes the faculty as a whole expects from graduates from the program. While you may choose as many outcomes as you wish, it is often a good strategy to focus on the most important goals for students in the first few years of your Student Assessment Program. For example, two or three of the most critical goals would be a good starting point. Please complete questions 1-6 for each Student Learning Outcome you are assessing or plan to evaluate in the next review period on the attached sheet.
Please complete this page for each Student Learning Outcome (minimum of 3) for each of your degree programs.

Degree Program Title: __________________   Degree Type: _________________
Banner/CIP Code: ______________

STUDENT LEARNING OUTCOME # 1

1. STUDENT LEARNING OUTCOME (SLO): What will a student be able to do, what knowledge, skills, values will they have, etc., upon graduation from the program that will be assessed? A Student Learning Outcome is a clear concise statement that describes how students can demonstrate their mastery of some element of the academic program goals.

Program Objective #1 (PO1): Students will develop skills in problem analysis.
Students will be able to do the following:

(CO270.2) Be able to use the linked list data structure.
(CO270.3) Be able to use the stack data structure.
(CO270.4) Be able to use the queue data structure.
(CO270.8) Be able to use the binary tree data structure and a hash table.

(CO340.6) Describe, define and apply the major components of the relational database model.
(CO340.8) Describe the fundamental data structures, access methods and storage devices needed for physical database design.

(CO359.2) Explain the purpose and activities of the systems development life cycle phases.

(CO380.1) Creation and manipulation of web graphics using popular software tools.
(CO380.2) Creation of Web Pages using XHTML
(CO380.3) Application of cascading style sheets

(CO428.1) Have an understanding of Software Engineering Basics.
(CO428.6) Be able to utilize UML for software development purposes.

(CO431.9) Be able to use the JAVA Swing library to develop programs with graphical user interfaces.

CSCI 440 Applied Software Project Development
(CO440.6) Build user-friendly, aesthetic, and functional interfaces for application software projects.
(CO440.8) Develop and implement a system application project in an object-oriented programming language using traditional process model diagrams as a guide.

(CO470.1) Identify and explain the major components of the relational data model.
(CO470.2) Utilize structured query language (SQL) to define and manipulate database objects in the interactive mode.
(CO470.3) Incorporate procedural extensions to SQL for maintaining database tables.
(CO470.4) Develop an application program to access databases with the Java programming language.
Perform system and database administration to implement software to support database application development.

Complete a project to implement database management software or related tools.

2. **LINKS TO CURRICULUM & PROGRAM FACULTY.** *What courses support this SLO? How do all program faculty participate in setting the goals, content and learning outcomes of these courses? How do all program faculty participate in analyzing and making recommendations based on the results of student assessments?*

Program Objective #1 (PO1): Students will develop skills in problem analysis.

Program Objective(PO)#1 is evaluated by scoring the following course objectives (CO) from:

**CSCI 270**
(CO270.2) Be able to use the linked list data structure.
(CO270.3) Be able to use the stack data structure.
(CO270.4) Be able to use the queue data structure.
(CO270.8) Be able to use the binary tree data structure and a hash table.

**CSCI 340 Introduction to Database Systems**
(CO340.6) Describe, define and apply the major components of the relational database model.
(CO340.8) Describe the fundamental data structures, access methods and storage devices needed for physical database design.

**CSCI 359 Systems Analysis and Design**
(CO359.2) Explain the purpose and activities of the systems development life cycle phases.

**CSCI 380**
(CO380.1) Creation and manipulation of web graphics using popular software tools.
(CO380.2) Creation of Web Pages using XHTML
(CO380.3) Application of cascading style sheets

**CSCI 428 Object Oriented Programming**
(CO428.1) Software Engineering Basic.
(CO428.6) UML

**CSCI 431 JAVA Programming**
(CO431.9) Use the Swing library to develop programs with graphical user interfaces.

**CSCI 440 Applied Software Project Development**
(CO440.6) Build user-friendly, aesthetic, and functional interfaces for application software projects.
(CO440.8) Develop and implement a system application project in an object-oriented programming language using traditional process model diagrams as a guide.
Each faculty member, who teaches a particular course, is responsible for developing a criteria for evaluating each course objectives (CO). The faculty member will report a percentile expression of success for each course objective in each course he or she teaches. If the percentile is less than 75%, the instructor will give a brief explanation of the reason for the low percentile and suggest a possible remedy. Any course objective below 75% is a candidate for an action item.
3. **STUDENT LEARNING OUTCOME (SLO)**: What will a student be able to do, what knowledge, skills, values will they have, etc., upon graduation from the program that will be assessed? A Student Learning Outcome is a clear concise statement that describes how students can demonstrate their mastery of some element of the academic program goals.

**Program Objective #2 (PO2): Students will develop problem-solving skills.**

Students will be able to do the following:

(CO152.1) Be able to use one-dimensional arrays.
(CO152.2) Be able to use at least one (preferably at least two) sorting technique(s) to rearrange data in an array.
(CO152.3) Be able to search an array using both linear and binary searching techniques.
(CO152.7) Be able to design and code a program which includes a user-created class.

(CO241.2) Be able to understand concepts of Machine Instructions, Assembly and linking, assembly language programming (Unconditional jumps, flags, subroutines, Stacks)

(CO270.1) Be able to use address variables.
(CO270.8) Be able to integrate the use of container classes (user-created or STL) into a moderately complex program solution.

(CO431.7) Be able to employ exception-handling programming techniques.
(CO431.8) Be able to utilize file input and output procedures for sequential and random access.
(CO431.9) Be able to use the JAVA Swing library to develop programs with graphical user interfaces.

(CO440.9) Be able to connect a database and interface to software project.

(CO470.4) Be able to develop an application program to access databases with the Java programming language.

4. **LINKS TO CURRICULUM & PROGRAM FACULTY.** What courses support this SLO? How do all program faculty participate in setting the goals, content and learning outcomes of these courses? How do all program faculty participate in analyzing and making recommendations based on the results of student assessments?

**Program Objective #2 (PO2): Students will develop problem-solving skills.**

Assessment will be measured through testing the following course objectives:

The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 152 Programming Fundamentals II
(CO152.1) Be able to use one-dimensional arrays.
(CO152.2) Be able to use at least one (preferably at least two) sorting technique(s) to rearrange data in an array.
(CO152.3) Be able to search an array using both linear and binary searching techniques.
(CO152.7) Be able to design and code a program which includes a user-created class.

CSCI 241 Assembly Language and Computer Organization
(CO241.2) Concepts of Machine Instructions, Assembly and linking, assembly language programming (Unconditional jumps, flags, subroutines, Stacks)

CSCI 270 Data Structure and Algorithms
(CO270.1) Be able to use address variables.
(CO270.8) Be able to integrate the use of container classes (user-created or STL) into a moderately complex program solution.

CSCI 431 JAVA Programming
(CO431.7) Employ exception-handling programming techniques.
(CO431.8) Utilize file input and output procedures for sequential and random access.
(CO431.9) Use the Swing library to develop programs with graphical user interfaces.

CSCI 440 Applied Software Project Development
(CO440.9) Connect a database and interface to software project.

CSCI 470 Database Programming
(CO470.4) Develop an application program to access databases with the Java programming language.

Each faculty member, who teaches a particular course, is responsible for developing a criteria for evaluating each course objectives (CO). The faculty member will report a percentile expression of success for each course objective in each course he or she teaches. If the percentile is less than 75%, the instructor will give a brief explanation of the reason for the low percentile and suggest a possible remedy. Any course objective below 75% is a candidate for an action item.
5. **STUDENT LEARNING OUTCOME (SLO):** What will a student be able to do, what knowledge, skills, values will they have, etc., upon graduation from the program that will be assessed? A Student Learning Outcome is a clear concise statement that describes how students can demonstrate their mastery of some element of the academic program goals.

**Program Objective #3 (PO3): Students will develop solution-modeling skills.**

Students will be able to do the following:

(CO340.1) Model a single entity, define and access a single entity database.
(CO340.2) Model a one-to-many (1:m) relationship between two entities, define a 1:m database, and process a 1:m database.
(CO340.3) Model a m:m relationship between two entities, define and process a m:m database.
(CO340.4) Create a well-formed, high fidelity data model.
(CO340.5) Describe the process of normalization and distinguish between different normal forms.

(CO359.5) Understand and model system entities and data stores.
(CO359.6) Understand and model system processes, events, and data flows within a system.
(CO359.7) Understand and model classes of data within a system.
(CO359.8) Understand concepts relating to various models, tools, and techniques used in system analysis and design.

(CO440.2) Use Microsoft Visio to create, edit, and publish to a web site traditional process model diagrams.
(CO440.3) Use Microsoft Visio to create, edit, and publish to a web site Entity-Relationship diagrams.
(CO440.7) Create a database using an Entity-Relationship diagram.

6. **LINKS TO CURRICULUM & PROGRAM FACULTY.** What courses support this SLO? How do all program faculty participate in setting the goals, content and learning outcomes of these courses? How do all program faculty participate in analyzing and making recommendations based on the results of student assessments?

**Program Objective #3 (PO3): Students will develop solution-modeling skills.**

CSCI 340 Introduction to Database Systems
(CO340.1) Model a single entity, define and access a single entity database.
(CO340.2) Model a one-to-many (1:m) relationship between two entities, define a 1:m database, and process a 1:m database.
(CO340.3) Model a m:m relationship between two entities, define and process a m:m database.
(CO340.4) Create a well-formed, high fidelity data model.
(CO340.5) Describe the process of normalization and distinguish between different normal forms.

CSCI 359 Systems Analysis and Design
(CO359.5) Understand and model system entities and data stores.
(CO359.6) Understand and model system processes, events, and data flows within a system.
(CO359.7) Understand and model classes of data within a system.
(CO359.8) Understand concepts relating to various models, tools, and techniques used in system analysis and design.

CSCI 440 Applied Software Project Development
(CO440.2) Use Microsoft Visio to create, edit, and publish to a web site traditional process model diagrams.
(CO440.3) Use Microsoft Visio to create, edit, and publish to a web site Entity-Relationship diagrams.
(CO440.7) Create a database using an Entity-Relationship diagram.

Each faculty member, who teaches a particular course, is responsible for developing a criteria for evaluating each course objectives (CO). The faculty member will report a percentile expression of success for each course objective in each course he or she teaches. If the percentile is less than 75%, the instructor will give a brief explanation of the reason for the low percentile and suggest a possible remedy. Any course objective below 75% is a candidate for an action item.
1. **STUDENT LEARNING OUTCOME (SLO):** What will a student be able to do, what knowledge, skills, values will they have, etc., upon graduation from the program that will be assessed? A Student Learning Outcome is a clear concise statement that describes how students can demonstrate their mastery of some element of the academic program goals. 

**Program Objective #4 (PO4): Students will develop solution-implementation skills.**

Students will be able to do the following:

- (CO152.4) Be able to use multiple-dimensional arrays.
- (CO152.5) Be able to use structs.
- (CO152.6) Be able to use classes.

- (CO241.2) Be able to understand concepts of Machine Instructions, Assembly and linking, assembly language programming (Unconditional jumps, flags, subroutines, Stacks)
- (CO241.4) Be able to understand concepts of I/O devices; memory mapped I/O; Interrupts; Arrays, addressing modes and Floating Point Instructions

- (CO270.5) Be able to design, code, and use recursive functions.

- (CO359.4) Be able to identify and understand system inputs and outputs.

- (CO340.7) Be able to understand and apply the Structured Query Language (SQL) for database definition and manipulation.
- (CO340.9) Be able to develop a procedural language application program to update a database table.

- (CO380.1) Be able to create and manipulate web graphics using popular software tools.
- (CO380.2) Be able to create Web Pages using XHTML
- (CO380.3) Be able to apply cascading style sheets
- (CO380.4) Be able to apply Client Side Scripting using JavaScript
- (CO380.5) Be able to implement Database creation and Web Integration using server side scripting.
- (CO380.6) Be able to utilize Ajax and Web 2.0 technologies to create Rich Internet Applications

- (CO431.1) Be able to code, compile and run a Java program.
- (CO431.2) Be able to master programming techniques for console input and output.
- (CO431.3) Be able to apply logical constructs for branching and loops.
- (CO431.7) Be able to employ exception-handling programming techniques.
- (CO431.8) Be able to utilize file input and output procedures for sequential and random access.
- (CO431.9) Be able to use the Swing library to develop programs with graphical user interfaces.

- (CO440.1) Be able to develop and maintain an informational and project repository web site for an application project.
(CO470.2) Be able to utilize structured query language (SQL) to define and manipulate database objects in the interactive mode.

(CO470.5) Be able to design a database-supported Web site.

(CO470.6) Be able to develop a database-supported Web site utilizing HTML and JavaServer Pages.

(CO470.7) Be able to apply XML for Data Exchange.

2. **LINKS TO CURRICULUM & PROGRAM FACULTY.** What courses support this SLO? How do all program faculty participate in setting the goals, content and learning outcomes of these courses? How do all program faculty participate in analyzing and making recommendations based on the results of student assessments?

**Program Objective #4 (PO4): Students will develop solution-implementation skills.**

Assessment will be measured through testing the following course objectives:

The first percentile is Fall 2009 and the second percentile is Spring 2010.

**CSCI 152 Programming Fundamentals II**
- (CO152.4) Be able to use multiple-dimensional arrays.
- (CO152.5) Be able to use structs.
- (CO152.6) Be able to use classes.

**CSCI 241 Machine Language and Computer Organization**
- (CO241.2) Concepts of Machine Instructions, Assembly and linking, assembly language programming (Unconditional jumps, flags, subroutines, Stacks)
- (CO241.4) I/O devices; memory mapped I/O; Interrupts; Arrays, addressing modes and Floating Point Instructions

**CSCI 270**
- (CO270.5) Be able to design, code, and use recursive functions.

**CSCI 359 Systems Analysis and Design**
- (CO359.4) Identify and understand system inputs and outputs.

**CSCI 340 Introduction to Database Systems**
- (CO340.7) Learn and apply the Structured Query Language (SQL) for database definition and manipulation.
- (CO340.9) Develop a procedural language application program to update a database table.

**CSCI 380 Web Programming and Interface Design**
- (CO380.1) Creation and manipulation of web graphics using popular software tools.
- (CO380.2) Creation of Web Pages using XHTML
- (CO380.3) Application of cascading style sheets
- (CO380.4) Client Side Scripting using JavaScript
- (CO380.5) Database creation and Web Integration using server side scripting.
(CO380.6) Utilize Ajax and Web 2.0 technologies to create Rich Internet Applications

CSCI 431 JAVA Programming
(CO431.1) Code, compile and run a Java program.
(CO431.2) Master programming techniques for console input and output.
(CO431.3) Apply logical constructs for branching and loops.
(CO431.7) Employ exception-handling programming techniques.
(CO431.8) Utilize file input and output procedures for sequential and random access.
(CO431.9) Use the Swing library to develop programs with graphical user interfaces.

CSCI 440 Applied Software Project Development
(CO440.1) Develop and maintain an informational and project repository web site for an application project.

CSCI 470 Database Programming
(CO470.2) Utilize structured query language (SQL) to define and manipulate database objects in the interactive mode.
(CO470.5) Design a database-supported Web site.
(CO470.6) Develop a database-supported Web site utilizing HTML and JavaServer Pages.
(CO470.7) Apply XML for Data Exchange.

Each faculty member, who teaches a particular course, is responsible for developing a criteria for evaluating each course objectives (CO). The faculty member will report a percentile expression of success for each course objective in each course he or she teaches. If the percentile is less than 75%, the instructor will give a brief explanation of the reason for the low percentile and suggest a possible remedy. Any course objective below 75% is a candidate for an action item.
STUDENT LEARNING OUTCOME # 5

1. STUDENT LEARNING OUTCOME (SLO): What will a student be able to do, what knowledge, skills, values will they have, etc., upon graduation from the program that will be assessed? A Student Learning Outcome is a clear concise statement that describes how students can demonstrate their mastery of some element of the academic program goals.

Program Objective #5 (PO5) : Students will develop ethics and strong communication skills.

Students will be able to do the following:

(CO251.1) Be able to define ethics, morality, and moral system and recognize the distinction between ethical theory and professional ethics.
(CO251.2) Be able to summarize the basic concepts of relativism, utilitarianism, and deontological theories.
(CO251.3) Be able use methods and tools of analysis to analyze an argument to identify premises and conclusion and illustrate the use of example, analogy, and counter-analogy in an ethical argument.
(CO251.4) Be able identify the strengths and weaknesses of relevant professional codes as expressions of professionalism and guides to decision-making.
(CO251.5) Be able summarize the legal bases for the right to privacy and freedom of expression in one's own nation and how those concepts vary from country to country.
(CO251.6) Be able identify the professional's role in security and the tradeoffs involved.
(CO251.7) Be able outline the technical basis of viruses and denial-of-service attacks and enumerate techniques to combat the same.
(CO251.8) Be able distinguish among patent, copyright, and trade secret protection and explain how patent and copyright laws may vary internationally.
(CO251.9) Be able explain the various U.S. legislation and regulations that impact technology and the disadvantages and advantages of free expression in cyberspace.
(CO251.10) Be able explain why computing/network access is restricted in some countries.
(CO251.11) Be able define a computer use policy with enforcement measures.

(CO359.3) Be able to identify and understand project management techniques.

(CO440.4) Be able develop and use a team constitution.
(CO440.5) Be able solve team conflicts in a project building environment.
(CO440.10) Be able create system documentation including help files, diagrams, and programming code.
(CO440.11) Be able present the final project to an audience consisting of faculty, peers, administrators, and business leaders.
(CO440.12) Be able evaluate other team members based upon specific criteria. (Derived based on team member evaluations.)

2. LINKS TO CURRICULUM & PROGRAM FACULTY. What courses support this SLO? How do all program faculty participate in setting the goals, content and learning outcomes of these courses? How do all program faculty participate in analyzing and making recommendations based on the results of student assessments?

Program Objective #5 (PO5) : Students will develop ethics and strong communication skills.
Assessment will be measured through testing the following course objectives:

CSCI 251 Introduction to Information Security, Law, and Ethics
(CO251.1) Define ethics, morality, and moral system and recognize the distinction between ethical theory and professional ethics.
(CO251.2) Summarize the basic concepts of relativism, utilitarianism, and deontological theories.
(CO251.3) Use methods and tools of analysis to analyze an argument to identify premises and conclusion and illustrate the use of example, analogy, and counter-analogy in an ethical argument.
(CO251.4) Identify the strengths and weaknesses of relevant professional codes as expressions of professionalism and guides to decision-making.
(CO251.5) Summarize the legal bases for the right to privacy and freedom of expression in one’s own nation and how those concepts vary from country to country.
(CO251.6) Identify the professional’s role in security and the tradeoffs involved.
(CO251.7) Outline the technical basis of viruses and denial-of-service attacks and enumerate techniques to combat the same.
(CO251.8) Distinguish among patent, copyright, and trade secret protection and explain how patent and copyright laws may vary internationally.
(CO251.9) Explain the various U.S. legislation and regulations that impact technology and the disadvantages and advantages of free expression in cyberspace.
(CO251.10) Explain why computing/network access is restricted in some countries.
(CO251.11) Define a computer use policy with enforcement measures.

CSCI 359 Systems Analysis and Design
(CO359.3) Understand project management techniques.

CSCI 440 Applied Software Project Development
(CO440.4) Develop and use a team constitution.
(CO440.5) Solve team conflicts in a project building environment.
(CO440.10) Create system documentation including help files, diagrams, and programming code.
(CO440.11) Present the final project to an audience consisting of faculty, peers, administrators, and business leaders.
(CO440.12) Evaluate other team members based upon specific criteria. (Derived based on team member evaluations.)

Each faculty member, who teaches a particular course, is responsible for developing a criteria for evaluating each course objectives (CO). The faculty member will report a percentile expression of success for each course objective in each course he or she teaches. If the percentile is less than 75%, the instructor will give a brief explanation of the reason for the low percentile and suggest a possible remedy. Any course objective below 75% is a candidate for an action item.
STUDENT LEARNING OUTCOME # 6

1. **STUDENT LEARNING OUTCOME (SLO):** What will a student be able to do, what knowledge, skills, values will they have, etc., upon graduation from the program that will be assessed? A Student Learning Outcome is a clear concise statement that describes how students can demonstrate their mastery of some element of the academic program goals.

**Program Objective #6 (PO6) : Learn common algorithms and how to analyze them for efficiency.**
Students will be able to do the following:

- (CO152.7) Be able to design and code a program which includes a user-created class.

- (CO270.6) Understand Big-O notation (for algorithm efficiency): what it means, how it is determined, and why it should be considered in effective programming.

- (CO270.7) Be able to use the binary tree data structure and a hash table.

2. **LINKS TO CURRICULUM & PROGRAM FACULTY.** What courses support this SLO? How do all program faculty participate in setting the goals, content and learning outcomes of these courses? How do all program faculty participate in analyzing and making recommendations based on the results of student assessments?

**Program Objective #6 (PO6) : Learn common algorithms and how to analyze them for efficiency.**
CSCI 152
- (CO152.7) Be able to design and code a program which includes a user-created class.

CSCI 270
- (CO270.6) Understand Big-O notation (for algorithm efficiency): what it means, how it is determined, and why it should be considered in effective programming.
- (CO270.7) Be able to use the binary tree data structure and a hash table.

Each faculty member, who teaches a particular course, is responsible for developing a criteria for evaluating each course objectives (CO). The faculty member will report a percentile expression of success for each course objective in each course he or she teaches. If the percentile is less than 75%, the instructor will give a brief explanation of the reason for the low percentile and suggest a possible remedy. Any course objective below 75% is a candidate for an action item.
STUDENT LEARNING OUTCOME # 7

1. STUDENT LEARNING OUTCOME (SLO): What will a student be able to do, what knowledge, skills, values will they have, etc., upon graduation from the program that will be assessed? A Student Learning Outcome is a clear concise statement that describes how students can demonstrate their mastery of some element of the academic program goals.

Program Objective #7 (PO7) : Learn theory behind modern computer technologies.

Students will be able to do the following:

- Be able to utilize various numbering systems and conversions.
- Understand Computer Organization concepts: registers, transfers, machine cycles.
- Understand concepts of I/O devices, memory mapped I/O; Interrupts.
- Understand Software Engineering Basics.
- Understand Classes basic and advanced.
- Be able to utilize UML in planning software.
- Understand the concepts, structures, and mechanisms of operating systems.
- Understand memory management, virtual memory, swapping, paging algorithms, segmentation, and clock paging policies.
- Understand multiprogramming and multiuser capabilities, and how operating systems evolved.
- Understand process management, process states and process and thread structures and concepts.
- Understand concurrent processes and associated deadlock prevention, avoidance, detection, recovery methods, and the use of semaphores.
- Learn specific design decisions and architectures used in modern operating systems.
- Understand concepts relating to different types of information systems.

2. LINKS TO CURRICULUM & PROGRAM FACULTY. What courses support this SLO? How do all program faculty participate in setting the goals, content and learning outcomes of these courses? How do all program faculty participate in analyzing and making recommendations based on the results of student assessments?

Program Objective #7 (PO7) : Learn theory behind modern computer technologies.

Assessment will be measured through testing the following course objectives:

The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 241
- (CO241.1) Understand various numbering systems and conversions.
- (CO241.3) Understand Computer Organization: registers, transfers, machine cycles.
- (CO241.4) Understand I/O devices, memory mapped I/O; Interrupts.

CSCI 428 Object Oriented Programming
- (CO428.1) Software Engineering Basic.
Each faculty member, who teaches a particular course, is responsible for developing a criteria for evaluating each course objectives (CO). The faculty member will report a percentile expression of success for each course objective in each course he or she teaches. If the percentile is less than 75%, the instructor will give a brief explanation of the reason for the low percentile and suggest a possible remedy. Any course objective below 75% is a candidate for an action item.

7. **ACTION PLAN: STRATEGIES/METHODS FOR OBSERVING STUDENT LEARNING.** How will data be collected, analyzed, shared? How will faculty observe the accomplishment of this outcome? Please provide specific descriptions for how, when, how often, what course(s), what student performances will be observed, collected and analyzed. Please provide or attach any descriptions of your ACTION PLAN OR PROCESS addressing the who, what, when, where questions for the assessment program.

Course objectives are established by the entire faculty. These course objectives cannot be changed without the consideration and permission of the Computer Science faculty. However, each instructor can decide on how each Course Objective will be evaluated. This evaluation is in the form of a percentile. Below is an description of how each faculty member derives the success percentile for each course he or she teaches.

Course: CSCI 151.003 Fall 2010

Professor: Thomas L. Brown

1. Construct appropriate comments .
2. Declare valid identifiers using appropriate data types.
3. Input and output data.
4. Evaluate and construct selection structures.
5. Evaluate and construct repetition structures.
6. Construct programs using multiple functions.*
7. Understand the concepts of scope and lifetime.*
8. Understand how and why to use value and reference parameters with functions.*
9. Effectively use one-dimensional arrays.*

Derivation of Assessment Scores:
#1 from exam 1
#2 from quiz 2
#3 from lab assignment 2
#4 from quiz 3
#5 from exam 2
#6 from lab assignment 7
#7 from quiz 4
#8 from lab assignment 9
#9 from exam 3

Course: CSCI 152.001 Programming Fundamentals II Fall 2010
Instructor: Sandy Huerter

152 Course Objectives

1) Be able to use one-dimensional arrays and strings.
2) Be able to use at least one (preferably at least two) sorting technique(s) to rearrange data in an array.
3) Be able to search an array using both linear and binary searching techniques.
4) Be able to use multiple-dimensional arrays.
5) Be able to use structs.
6) Be able to create and use classes.
7) Be able to design and code a program which includes a user-created class.

Derivation of Assessment Scores:

#1 based on quiz 2, final exam
#2 based on quiz 4, final exam
#3 based on quiz 4, final exam
#4 based on quiz 3, final exam
#5 based on quiz 4, final exam
#6 based on final exam
#7 based on final exam, program 5

Course: CSCI 233.001 Fall 2010
Instructor: Thomas L. Brown
1. Compile and test a program.
2. Design and develop a basic report program.
3. Enhance a basic report program to process grouped data and summarize results.
4. Learn the programming constructs and develop programs to create and process arrays.
5. Develop a program to capture, process and store object data (class instance) into a file.*
6. Design and develop a program to process a sequentially-organized file.*
7. Develop a program to access data from a database.
8. Design and develop a basic input form to capture data for an application.
9. Design and develop pages for a basic online application (paired programming)*.

Derivation of Assessment Scores:
#1 from exercises 1
#2 from exercise 2 & 3
#3 from exercise 4
#4 from exam 5
#5 from exercise 6
#6 from exercises 7
#7 from exercise 8
#8 from exercises 9 & 10
#9 from project

CSCI 241 Machine Language and Computer Organization
Instructor: Sam Saffer, Ph.D.

Course Objectives: Students will gain knowledge and understandings of the following:

(CO241.1) Binary numbering systems and conversions; floating point representation
(CO241.2) Concepts of Machine Instructions, Assembly and linking, assembly language programming (Unconditional jumps, flags, subroutines, Stacks)
(CO241.3) Intro to Computer Organization
(CO241.4) I/O devices; memory mapped I/O; Interrupts ; Arrays, addressing modes and Floating Point Instructions.
(CO241.5) Integration of assembly language instructions, machine cycles, and computing organization into an understanding of how modern computer hardware functions.

Objective #1 – Test #1
Objective #2 – Test #2
Objective #3 – Test #3
Objective #4 – Test #4
Objective #5 – Final Exam
Course: CSCI 251.01W Introduction to Information Security, Law, and Ethics Fall 2010
Professor: Will McWhorter

1. Define ethics, morality, and moral system and recognize the distinction between ethical theory and professional ethics
2. Summarize the basic concepts of relativism, utilitarianism, and deontological theories.
3. Use methods and tools of analysis to analyze an argument to identify premises and conclusion and illustrate the use of example, analogy, and counter-analogy in an ethical argument.
4. Identify the strengths and weaknesses of relevant professional codes as expressions of professionalism and guides to decision-making.
5. Summarize the legal bases for the right to privacy and freedom of expression in one’s own nation and how those concepts vary from country to country.
6. Identify the professional’s role in security and the tradeoffs involved.
7. Outline the technical basis of viruses and denial-of-service attacks and enumerate techniques to combat the same.
8. Distinguish among patent, copyright, and trade secret protection and explain how patent and copyright laws may vary internationally.
9. Explain the various U.S. legislation and regulations that impact technology and the disadvantages and advantages of free expression in cyberspace.
10. Explain why computing/network access is restricted in some countries.
11. Define a computer use policy with enforcement measures.

Derivation of Assessment Scores:
#1 based on Midterm Exam and Quiz 1
#2 based on Midterm, Final Exam, and Quiz 2
#3 based on Midterm Exam and Quiz 3
#4 based on Midterm, Final Exam, and Quiz 4
#5 based on Midterm Exam and Quiz 5
#6 based on Midterm Exam and Quiz 6
#7 based on Final Exam and Quiz 7
#8 based on Midterm, Final Exam, and Quiz 8
#9 based on Midterm, Final Exam, and Quiz 9
#10 based on Final Exam and Quiz 10
#11 based on Final Exam and Quiz 11

Overall Assessment of Objectives
Course: CSCI 270.001 Data Structures Fall 2010
Instructor: Sandy Huerter

1) Be able to use address variables.
2) Be able to use the linked list data structure.
3) Be able to use the stack data structure.
4) Be able to use the queue data structure.
5) Be able to design, code, and use recursive functions.
6) Understand Big-O notation (for algorithm efficiency): what it means, how it is determined, and why it should be considered in effective programming.
7) Be able to use the binary tree data structure and a hash table.
8) Be able to integrate the use of container classes (user-created or STL) into a moderately complex program solution.

Derivation of Assessment Scores:
#1 based on exam 2
Course: CSCI 340.001 Fall 2010  
Instructor: Thomas L. Brown  
1. Model a single entity, define and access a single entity database  
2. Model a one-to-many (1:m) relationship between two entities, define a 1:m database, and process a 1:m database.  
3. Model a m:m relationship between two entities, define and process a m:m database.*  
4. Create a well-formed, high fidelity data model.  
5. Describe the process of normalization and distinguish between different normal forms.  
6. Describe, define and apply the major components of the relational database model.  
7. Learn and apply the Structured Query Language (SQL) for database definition and manipulation.  
8. Describe the fundamental structures, access methods and other components needed for database design.  
9. Develop a procedural language application program to update a database table.*  

Derivation of Assessment Scores:  
#1 assignment 1, 4  
#2 assignment 7-9  
#3 assignment 10-13  
#4 assignments 1-3  
#5 final exam  
#6 assignments 2, 5  
#7 midterm exam  
#8 assignments 16, 17  
#9 assignments 18-20  
* denotes unsuccessful objectives (< 75%)  

CSCI 434 Introduction to Local Area Networks  
Instructor: S. Saffer, Ph.D.  
Objective #1: To define and understand basic Data Communications (common terms, network topologies, networking media, physical and logical topologies).  
Objective #2: To understand networking topologies, the OSI Model and the IEEE 802 standards, 9802.3, 802.4, 802.5, 802.11).  
Objective #3: To gain practical experience with subnetting, and the use of TCP/IP, IP addresses, and the fundamentals of IP routing.  
Objective #4: To gain exposure to various networking platforms within the SPX/IPX and TCP/IP environment; To gain an overall understanding of local area
networking technology.

Measurement:
Objection #1 is measured by Exam #1
Objection #2 is measured by Exam #2
Objection #3 is measured by Exam #3
Objection #4 is measured by the Final Exam
8. **CRITERIA FOR SUCCESS: MEASURES & TARGETS.** What are the standards of progress or criteria used for judging success for the student learning assessment observations? Please attach any assessment tools, standards (rubrics) or other documents used to judge success or achievement of the outcome.

These two additional reports for questions 5&6 below will be due in May 11, 2012

5. **ACHIEVEMENT SUMMARY: FINDINGS & RESULTS.** What are the results of the assessment of this learning objective thus far? Be sure to include the year of the assessment, attach any relevant reports, data tables, etc. Please be specific in your descriptions. Indicating that n% students took a test or passed an oral exam is not an example of assessment findings.

6. **PROGRAM ENHANCEMENT.** How has assessment data been used? Please give examples over the last 3 years. What are the specific mechanisms for communicating results and changing courses, curriculum, learning activities within a course, etc?

Review and Approval Signatures & Date:
Program Coordinator if applicable ________________________________
Department Chair: ________________________________
Dean ________________________________

5&6 (See below)
These two additional reports for questions 5&6 below will be due in May 11, 2012

5. ACHIEVEMENT SUMMARY: FINDINGS & RESULTS. What are the results of the assessment of this learning objective thus far? Be sure to include the year of the assessment, attach any relevant reports, data tables, etc. Please be specific in your descriptions. Indicating that n% students took a test or passed an oral exam is not an example of assessment findings.

6. PROGRAM ENHANCEMENT. How has assessment data been used? Please give examples over the last 3 years. What are the specific mechanisms for communicating results and changing courses, curriculum, learning activities within a course, etc.

5. ACHIEVEMENT SUMMARY: FINDINGS & RESULTS.

Assessment for the CS Degree Program 2010-2011
Fall 10 - Spring 11
Computer Science Dept.
Texas A&M University - Commerce

82%  83% Program Objective #1 (PO1): Students will develop skills in problem analysis.
80%  82% Program Objective #2 (PO2): Students will develop problem-solving skills.
79%  85% Program Objective #3 (PO3): Students will develop solution-modeling skills.
82%  79% Program Objective #4 (PO4): Students will develop solution-implementation skills.
87%  86% Program Objective #5 (PO5): Students will develop strong communication skills.
78%  80% Program Objective #6 (PO6): Learn common algorithms and how to analyze them for efficiency.
88%  89% Program Objective #7 (PO7): Understand the concepts used in modern computer technologies.

Outcome Description
82%  83% Program Objective #1 (PO1): Students will develop skills in problem analysis.

Assessment Method
Assessment will be measured through testing the following course objectives:
The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 270
77%  77% (CO270.2) Be able to use the linked list data structure.
80%  80% (CO270.3) Be able to use the stack data structure.
79%  77% (CO270.4) Be able to use the queue data structure.
80%  86% (CO270.8) Be able to use the binary tree data structure and a hash table.

CSCI 340 Introduction to Database Systems
79%  79% (CO340.6) Describe, define and apply the major components of the relational database model.
76%  76% (CO340.8) Describe the fundamental data structures, access methods and storage devices needed for physical database design.

CSCI 359 Systems Analysis and Design <Data not available>
0% 0% (CO359.2) Explain the purpose and activities of the systems development life cycle phases.

CSCI 380
0% 0% (CO380.1) Creation and manipulation of web graphics using popular software tools.
0% 0% (CO380.2) Creation of Web Pages using XHTML
0% 0% (CO380.3) Application of cascading style sheets

CSCI 428 Object Oriented Programming
98% 100% (CO428.1) Software Engineering Basic.
85% 85% (CO428.6) UML

CSCI 431 JAVA Programming
0% 0% (CO431.9) Use the Swing library to develop programs with graphical user interfaces.

CSCI 440 Applied Software Project Development
0% 93% (CO440.6) Build user-friendly, aesthetic, and functional interfaces for application software projects.
0% 95% (CO440.8) Develop and implement a system application project in an object-oriented programming language using traditional process model diagrams as a guide.

CSCI 470 Database Programming
0% 80% (CO470.1) Identify and explain the major components of the relational data model.
0% 88% (CO470.2) Utilize structured query language (SQL) to define and manipulate database objects in the interactive mode.
0% 82% (CO470.3) Incorporate procedural extensions to SQL for maintaining database tables.
0% 82% (CO470.4) Develop an application program to access databases with the Java programming language.
0% 80% (CO470.8) Perform system and database administration to implement software to support database application development.
0% 75% (CO470.9) Complete a project to implement database management software or related tools.

**80% 82% Program Objective #2 (PO2): Students will develop problem-solving skills.**
Assessment will be measured through testing the following course objectives:
The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 152 Programming Fundamentals II
84% 76% (CO152.1) Be able to use one-dimensional arrays.
77% 80% (CO152.2) Be able to use at least one (preferably at least two) sorting technique(s) to rearrange data in an array.
76% 80% (CO152.3) Be able to search an array using both linear and binary searching techniques.
77% 79% (CO152.7) Be able to design and code a program which includes a user-created class.

CSCI 241 Assembly Language and Computer Organization
89% 84% (CO241.2) Concepts of Machine Instructions, Assembly and linking, assembly language programming (Unconditional jumps, flags, subroutines, Stacks)
CSCI 270 Data Structure and Algorithms
76% 76% (CO270.1) Be able to use address variables.
80% 86% (CO270.8) Be able to integrate the use of container classes (user-created or STL) into a moderately complex program solution.

CSCI 431 JAVA Programming
0% 0% (CO431.7) Employ exception-handling programming techniques.
0% 0% (CO431.8) Utilize file input and output procedures for sequential and random access.
0% 0% (CO431.9) Use the Swing library to develop programs with graphical user interfaces.

CSCI 440 Applied Software Project Development
0% 95% (CO440.9) Connect a database and interface to software project.

CSCI 470 Database Programming
0% 82% (CO470.4) Develop an application program to access databases with the Java programming language.

79% 85% Program Objective #3 (PO3): Students will develop solution-modeling skills.
Assessment will be measured through testing the following course objectives:
The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 340 Introduction to Database Systems
94% 92% (CO340.1) Model a single entity, define and access a single entity database.
83% 79% (CO340.2) Model a one-to-many (1:m) relationship between two entities, define a 1:m database, and process a 1:m database.
70% 72% (CO340.3) Model a m:m relationship between two entities, define and process a m:m database.
75% 75% (CO340.4) Create a well-formed, high fidelity data model.
72% 77% (CO340.5) Describe the process of normalization and distinguish between different normal forms.

CSCI 359 Systems Analysis and Design <Data not available>
0% 0% (CO359.5) Understand and model system entities and data stores.
0% 0% (CO359.6) Understand and model system processes, events, and data flows within a system.
0% 0% (CO359.7) Understand and model classes of data within a system.
0% 0% (CO359.8) Understand concepts relating to various models, tools, and techniques used in system analysis and design.

CSCI 440 Applied Software Project Development
0% 95% (CO440.2) Use Microsoft Visio to create, edit, and publish to a web site traditional process model diagrams.
0% 97% (CO440.3) Use Microsoft Visio to create, edit, and publish to a web site Entity-Relationship diagrams.
0% 96% (CO440.7) Create a database using an Entity-Relationship diagram.

82% 73% Program Objective #4 (PO4): Students will develop solution-implementation skills.
Assessment will be measured through testing the following course objectives:
The first percentile is Fall 2009 and the second percentile is Spring 2010.
CSCI 152 Programming Fundamentals II
81% 76% (CO152.4) Be able to use multiple-dimensional arrays.
83% 78% (CO152.5) Be able to use structs.
77% 77% (CO152.6) Be able to use classes.

CSCI 241 Machine Language and Computer Organization
89% 84% (CO241.2) Concepts of Machine Instructions, Assembly and linking, assembly language programming (Unconditional jumps, flags, subroutines, Stacks)
94% 85% (CO241.4) I/O devices; memory mapped I/O; Interrupts; Arrays, addressing modes and Floating Point Instructions

CSCI 270
80% 77% (CO270.5) Be able to design, code, and use recursive functions.

CSCI 359 Systems Analysis and Design
0% 0% (CO359.4) Identify and understand system inputs and outputs.

CSCI 340 Introduction to Database Systems
80% 80% (CO340.7) Learn and apply the Structured Query Language (SQL) for database definition and manipulation.
72% 62% (CO340.9) Develop a procedural language application program to update a database table.

CSCI 380 Web Programming and Interface Design
0% 0% (CO380.1) Creation and manipulation of web graphics using popular software tools.
0% 0% (CO380.2) Creation of Web Pages using XHTML
0% 0% (CO380.3) Application of cascading style sheets
0% 0% (CO380.4) Client Side Scripting using JavaScript
0% 0% (CO380.5) Database creation and Web Integration using server side scripting.
0% 0% (CO380.6) Utilize Ajax and Web 2.0 technologies to create Rich Internet Applications

CSCI 431 JAVA Programming
0% 0% (CO431.1) Code, compile and run a Java program.
0% 0% (CO431.2) Master programming techniques for console input and output.
0% 0% (CO431.3) Apply logical constructs for branching and loops.
0% 0% (CO431.7) Employ exception-handling programming techniques.
0% 0% (CO431.8) Utilize file input and output procedures for sequential and random access.
0% 0% (CO431.9) Use the Swing library to develop programs with graphical user interfaces.

CSCI 440 Applied Software Project Development
0% 90% (CO440.1) Develop and maintain an informational and project repository web site for an application project.

CSCI 470 Database Programming
0% 88% (CO470.2) Utilize structured query language (SQL) to define and manipulate database objects in the interactive mode.
0% 80% (CO470.5) Design a database-supported Web site.
0% 75% (CO470.6) Develop a database-supported Web site utilizing HTML and JavaServer Pages.
0% 0% (CO470.7) Apply XML for Data Exchange.
87% 86% Program Objective #5 (PO5) : Students will develop ethics and strong communication skills.
Assessment will be measured through testing the following course objectives:
The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 251 Introduction to Information Security, Law, and Ethics
83% 85% (CO251.1) Define ethics, morality, and moral system and recognize the distinction between ethical theory and professional ethics.
82% 86% (CO251.2) Summarize the basic concepts of relativism, utilitarianism, and deontological theories.
84% 78% (CO251.3) Use methods and tools of analysis to analyze an argument to identify premises and conclusion and illustrate the use of example, analogy, and counter-analogy in an ethical argument.
91% 84% (CO251.4) Identify the strengths and weaknesses of relevant professional codes as expressions of professionalism and guides to decision-making.
84% 85% (CO251.5) Summarize the legal bases for the right to privacy and freedom of expression in one’s own nation and how those concepts vary from country to country.
92% 83% (CO251.6) Identify the professional’s role in security and the tradeoffs involved.
87% 83% (CO251.7) Outline the technical basis of viruses and denial-of-service attacks and enumerate techniques to combat the same.
80% 76% (CO251.8) Distinguish among patent, copyright, and trade secret protection and explain how patent and copyright laws may vary internationally.
89% 83% (CO251.9) Explain the various U.S. legislation and regulations that impact technology and the disadvantages and advantages of free expression in cyberspace.
92% 86% (CO251.10) Explain why computing/network access is restricted in some countries.
90% 87% (CO251.11) Define a computer use policy with enforcement measures.

CSCI 359 Systems Analysis and Design
0% 0% (CO359.3) Understand project management techniques.

CSCI 440 Applied Software Project Development
0% 95% (CO440.4) Develop and use a team constitution.
0% 86% (CO440.5) Solve team conflicts in a project building environment.
0% 95% (CO440.10) Create system documentation including help files, diagrams, and programming code.
0% 93% (CO440.11) Present the final project to an audience consisting of faculty, peers, administrators, and business leaders.
0% 89% (CO440.12) Evaluate other team members based upon specific criteria. (Derived based on team member evaluations.)

78% 80% Program Objective #6 (PO6) : Learn common algorithms and how to analyze them for efficiency.
Assessment will be measured through testing the following course objectives:
The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 152
77% 79% (CO152.7) Be able to design and code a program which includes a user-created class.
75% 76% (CO270.6) Understand Big-O notation (for algorithm efficiency): what it means, how it is determined, and why it should be considered in effective programming.
81% 85% (CO270.7) Be able to use the binary tree data structure and a hash table.

88% 89% Program Objective #7 (PO7) : Learn theory behind modern computer technologies.
Assessment will be measured through testing the following course objectives:
The first percentile is Fall 2009 and the second percentile is Spring 2010.

CSCI 241
93% 89% (CO241.1) Understand various numbering systems and conversions.
76% 85% (CO241.3) Understand Computer Organization: registers, transfers, machine cycles.
94% 85% (CO241.4) Understand I/O devices, memory mapped I/O; Interrupts.

CSCI 428 Object Oriented Programming
98% 100% (CO428.1) Software Engineering Basic.
84% 88% (CO428.2) Classes basics/advanced
85% 85% (CO428.6) UML

CSCI 430 Operating Systems
0% 0% (CO430.1) Understand the concepts, structures, and mechanisms of operating systems.
0% 0% (CO430.2) Understand memory management, virtual memory, swapping, paging algorithms, segmentation, and clock paging policies.
0% 0% (CO430.3) Understand multiprogramming and multiuser capabilities, and how operating systems evolved.
0% 0% (CO430.4) Understand process management, process states and process and thread structures and concepts.
0% 0% (CO430.5) Understand concurrent processes and associated deadlock prevention, avoidance, detection, recovery methods, and the use of semaphores.
0% 0% (CO430.6) Learn specific design decisions and architectures used in modern operating systems.

CSCI 359 Systems Analysis and Design
0% 0% (CO359.1) Understand concepts relating to different types of information systems.
6. PROGRAM ENHANCEMENT. How has assessment data been used? Please give examples over the last 3 years. What are the specific mechanisms for communicating results and changing courses, curriculum, learning activities within a course, etc.

*How has assessment data been used?*

1. The assessment data is used as a mechanism to give each faculty member an overview of the operation of each degree program. Each semester, an assessment report is generated for each of the CS, CIS, and MS programs. Without this semester assessment data, it would be difficult for each faculty member, who has his or her own area of expertise and coursework, to be drawn into the involvement and overall functioning of the program. Consider the percentiles as gauges on an instrument panel. Each gauge may get a casual observance during the course of a given time period. However, the importance comes from the fact that a gauge reading below normal will get the deserved attention when something is malfunctioning.

2. The assessment data gives each degree program a topological structure, or hierarchical order, that allows each faculty member to better envision how the program is functioning. The fact that each course has a set of predetermined objectives, gives consistency to the course as Instructors changes over the semesters. Consistency is a particular problem when courses tend to be taught by adjuncts. The hierarchical order comes from the fact that various course objectives are used to evaluate a program objective. Thus the scoring of such Program Objective is dispersed over a number of courses and a number of Instructors.

Examples:
One of the most critical times for computer science students is at the very beginning when the students is learning the skills of computer programming. About half of the students have a difficult time with these beginning concepts. We have used assessment data of re-evaluate our beginning programming courses (CSCI 151, 152). We have used different teaching methodologies (such as the use of graphical programming) to make these courses more attractive to beginning students. We have noticed that if we can successfully interest a student interested in the "art" of computer programming at the CSCI 151 level, then that student will be more likely to successfully complete a computer science major. We are using the assessment data as we examine different teaching techniques as well as the use of different programming languages.