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

Student Learning Outcomes Check Sheet

Due 24 Feb 2012

Degree Program Title: MS 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 Master's Computer Science program, we have a 595 research course, which is usually taken in the final semester before graduation. In the course, students are required to design and implement a 1 semester project that captures the cumulative knowledge and accomplishments. The project can include a research project, as a continuation of current work with a faculty member. Involved in this course is computer programming, web page design, database and SQL subprojects, web database interface, and more. Furthermore, each student must pass the Comprehensive Exam, consisting of questions from CSCI 520 (Information Structure and Algorithm Analysis), CSCI 540 (Computer Architecture), CSCI 549 (Automata Theory), CSCI 530 (Operating Systems), CSCI 532 (Algorithm Design).
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:  _MS Computer Science_  
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 be able to demonstrate a broad knowledge of Computer Science which includes data structures, operating systems, computer programming skills, computer organization, algorithm design, and automata theory.

Students will be able to do the following:

(CO515.1): To understand the internal representation of the various data types.  
(CO515.2): To examine the internal representation of two and three dimension arrays in C/C++.  
(CO515.3): To understand dynamic memory allocation, parameter passing, the use of pointers.  

(CO516.1) Numbering systems and conversions, Boolean functions.  
(CO516.2) Intro to Computer Organization: design logic; digital diagrams, and basic circuits and gates, and the link between Boolean functions, circuits, processor and Micro code.  
(CO516.3) Concepts of Machine Instructions, Assembly and linking, common interrupts.  
(CO516.4): Concepts of Jumps, flags, subroutines, procedures, stacks, stack parameters and frames.  
(CO516.5): Arrays, addressing modes and Floating Point memory management, indirect addressing.  
(CO516.6): Advanced procedures, local variables, stack parameters, strings.  

(CO520.1): To understand the concept of sparse matrices, stack and queues.  
(CO520.2): To examine the differences between linear and linked representation of stacks, queues, and ordered data.  
(CO520.3): To understand and implement tree structures and to compare various sorting algorithms.

CSCI 530 Operating Systems
(CO530.1) Understand the concepts, structures, and mechanisms of operating systems.
(CO530.2) Understand memory management, virtual memory, swapping, paging algorithms, segmentation, and clock paging policies.
(CO530.5) Understand concurrent processes and associated deadlock prevention, avoidance, detection, recovery methods, and the use of semaphores.

(CO532.1): To teach students how to analyze algorithms in order to determine their calculation complexity in the terms of Big Oh, Big theta and Omega. Recursions.
(CO532.2): To teach sorting algorithms (such as mergesort and quicksort) and their applications.
(CO532.3): Probabilistic Analysis and Randomized algorithms for sample problems from the following list (not limited to, and not necessarily including all): CS- Hiring, Longest Streaks, Bins and Balls problem, the Birthday paradox, and randomized quicksort.
(CO532.4): Binary search trees and optimal binary search trees, and their applications.
(CO532.5): Dynamic programming algorithms for problems such as line scheduling, matrix chain multiplication, longest common subsequence, and their practical applications.
(CO532.6): Greedy algorithms for problems such as the activity selection problem and its application to resource planning.

(CO540.1): General purpose machines from different views. Instruction sets and classification of computers.
(CO540.3): Cache and memory organization: cache mapping and replace strategies, virtual memory and cache coherence.

(CO549.1): Understand the concept of languages and recursive definitions
(CO549.8): Construct a context free grammar to define a context free language
(CO549.11): Construct a push down automata for a language
(CO549.12): Design and construct a Turing machine for any language
(CO549.13): Design and construct a LR(1) parser for SmallG language

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 be able to demonstrate a broad knowledge of Computer Science which includes data structures, operating systems, computer programming skills, computer organization, algorithm design, and automata theory.

Assessment will be measured through testing the following course objectives: The first percentile is Fall 2010 and the second percentile is Spring 2011.

CSCI 515 Fundamental of Programming  
(CO515.1): To understand the internal representation of the various data types.  
(CO515.2): To examine the internal representation of two and three dimension arrays in C/C++.  
(CO515.3): To understand dynamic memory allocation, parameter passing, the use of pointers.

CSCI 516 Fundamental Concepts of Computer and Machine Organization  
(CO516.1) Numbering systems and conversions, Boolean functions.  
(CO516.2) Intro to Computer Organization: design logic; digital diagrams, and basic circuits and gates, and the link between Boolean functions, circuits, processor and Micro code.  
(CO516.3) Concepts of Machine Instructions, Assembly and linking, common interrupts.  
(CO516.4): Concepts of Jumps, flags, subroutines, procedures, stacks, stack parameters and frames..  
(CO516.5): Arrays, addressing modes and Floating Point memory management, indirect addressing.  
(CO516.6): Advanced procedures, local variables, stack parameters, strings.

CSCI 520 Information Structure and Algorithm Analysis.  
(CO520.1): To understand the concept of sparse matrices, stack and queues.  
(CO520.2): To examine the differences between linear and linked representation of stacks, queues, and ordered data.  
(CO520.3): To understand and implement tree structures and to compare various sorting algorithms.

CSCI 528 Object Oriented Methods  

CSCI 530 Operating Systems  
(CO530.1) Understand the concepts, structures, and mechanisms of operating systems.  
(CO530.2) Understand memory management, virtual memory, swapping, paging algorithms, segmentation, and clock paging policies.  
(CO530.5) Understand concurrent processes and associated deadlock prevention, avoidance, detection, recovery methods, and the use of semaphores.

CSCI 532 Algorithm Design  
(CO532.1): To teach students how to analyze algorithms in order to determine their calculation complexity in the terms of Big Oh, Big theta and Omega. Recursions.  
(CO532.2): To teach sorting algorithms (such as mergesort and quicksort) and their applications.
(CO532.3): Probabilistic Analysis and Randomized algorithms for sample problems from the following list (not limited to, and not necessarily including all): CS- Hiring, Longest Streaks, Bins and Balls problem, the Birthday paradox, and randomized quicksort.

(CO532.4): Binary search trees and optimal binary search trees, and their applications.

(CO532.5): Dynamic programming algorithms for problems such as line scheduling, matrix chain multiplication, longest common subsequence, and their practical applications.

(CO532.6): Greedy algorithms for problems such as the activity selection problem and its application to resource planning.

CSCI 540 Computer Architecture

(CO540.1): General purpose machines from different views. Instruction sets and classification of computers.


(CO540.3): Cache and memory organization: cache mapping and replace strategies, virtual memory and cache coherence.


CSCI 549 Automata Theory

(CO549.1): Understand the concept of languages and recursive definitions

(CO549.8): Construct a context free grammar to define a context free language

(CO549.11): Construct a push down automata for a language

(CO549.12): Design and construct a Turing machine for any language

(CO549.13): Design and construct a LR(1) parser for SmallG 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.
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 gain a substantial knowledge of one of the following Computer Science specialties: Database, Networking, Artificial Intelligence, Information Security, Computer Engineering. Students will be able to do the following:

(CO525.1): To define and understand basic Data Communications, networking topologies, the OSI Model and the IEEE 802 standards.

(CO526.1): Obtain current status of the state-of-the-art database design methodology in industry and academics.
(CO526.5): Write SQL programs for effective data definition and manipulation.
(CO526.6): Develop ER diagrams for logical design of database systems.
(CO526.7): Implement a small scale database development project using commercially available DBMS tools.

(CO534.1): Using subnets and routing protocols, design and configure a router network.
(CO534.2): Design and configure a switched network and VLANs.
(CO534.3): Understand the concepts of an Access Control List and learn how to configure a router for ACLs.

(CO538.1): To learn about general concepts in the field of artificial intelligence.
(CO538.2): To learn about the current fields of research in artificial intelligence.
(CO538.3): To work on an on-going class project to create a computer program that learns from its users.

(CO539.1): To learn about the general concepts and deployment of expert systems.
(CO539.2): To create an expert systems project using a pre-developed software tool (environment) or in any language of your choice.

(CO553.5): Become familiar with sockets, including programming both connection-oriented TCP and connectionless UDP sockets.
(CO553.6): Be able to create simple TCP Client/Server applications using sockets in a High-level language/toolbox such as Java, Perl, Python or C++.
(CO 563.1): State the basic concepts in information security, including security policies, security models, and various security mechanisms.
(.CO563.2): Understand the issues of network communications such as service, confidentiality, authentication, reliability, access control, and availability.
(.CO563.3): State threats and sources of attacks in network security.
(.CO563.4): Explain how to use cryptography to protect information and how to choose an appropriate encryption method.
(.CO563.5): State main strategies to secure Windows and Linux computers.
(.CO563.6): Understand limitation of the current security technology and able to choose proper security mechanisms.

(CO581.1): Students will be able to describe and discuss information security and network security basics.
(CO581.2): Students will be able to describe and discuss cryptography basics.
(CO581.3): Students will be able to describe and discuss authentication in network applications.
(CO581.4): Students will be able to describe and discuss electronic mail security.
(CO581.5): Students will be able to describe and discuss IP security.
(CO581.6): Students will be able to describe and discuss network security applications that implement the above capabilities.

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?

CSCI 525 Introduction to Local Area Networking
(CO525.1): To define and understand basic Data Communications, networking topologies, the OSI Model and the IEEE 802 standards.

CSCI 526 Databases Systems
(CO526.1): Obtain current status of the state-of-the-art database design methodology in industry and academics.
(CO526.5): Write SQL programs for effective data definition and manipulation.
(CO526.6): Develop ER diagrams for logical design of database systems.
(CO526.7): Implement a small scale database development project using commercially available DBMS tools.

CSCI 534 Networking II Routers and Switches
(CO534.1): Using subnets and routing protocols, design and configure a router network.
(CO534.2): Design and configure a switched network and VLANs.
(CO534.3): Understand the concepts of an Access Control List and learn how to configure a router for ACLs.

CSCI 538 Artificial Intelligence
(CO538.1): To learn about general concepts in the field of artificial intelligence.
(CO538.2): To learn about the current fields of research in artificial intelligence.
(CO538.3): To work on an on-going class project to create a computer program that learns from its users.
CSCI 539 Expert Systems
(CO539.1): To learn about the general concepts and deployment of expert systems.
(CO539.2): To create an expert systems project using a pre-developed software tool (environment) or in any language of your choice.

CSCI 553 Networking III – Unix Based Networks
(CO553.5): Become familiar with sockets, including programming both connection-oriented TCP and connectionless UDP sockets.
(CO553.6): Be able to create simple TCP Client/Server applications using sockets in a High-level language/toolbox such as Java, Perl, Python or C++.

CSCI 563 Fundamentals of Information Security & Assurance
(CO 563.1): State the basic concepts in information security, including security policies, security models, and various security mechanisms.
(CO563.2): Understand the issues of network communications such as service, confidentiality, authentication, reliability, access control, and availability.
(CO563.3): State threats and sources of attacks in network security.
(CO563.4): Explain how to use cryptography to protect information and how to choose an appropriate encryption method.
(CO563.5): State main strategies to secure Windows and Linux computers.
(CO563.6): Understand limitation of the current security technology and able to choose proper security mechanisms.

CSCI 581 Computer and Network Security
(CO581.1): Students will be able to describe and discuss information security and network security basics.
(CO581.2): Students will be able to describe and discuss cryptography basics.
(CO581.3): Students will be able to describe and discuss authentication in network applications.
(CO581.4): Students will be able to describe and discuss electronic mail security.
(CO581.5): Students will be able to describe and discuss IP security.
(CO581.6): Students will be able to describe and discuss network security applications that implement the above capabilities.

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 # 3

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.

Objective#3 (PO3): Students will demonstrate the ability to recognize, design and implement efficient software solutions to problems

Students will be able to do the following:

(CO520.1): To understand the concept of sparse matrices, stack and queues.
(CO520.2): To examine the differences between linear and linked representation of stacks, queues, and ordered data.
(CO520.3): To understand and implement tree structures and to compare various sorting algorithms.

(CO527.1): Understand current status of the state-of-the-art data mining methodology in industry and academics.
(CO527.3): Learn and use effective tools for web navigation and program integration management.
(CO527.5): Construct programs for capturing association rules.
(CO527.6): Write programs for trend analysis using statistical data mining techniques.

(CO528.2): Classes basics/advanced.
(CO528.3): Overloading.
(CO528.4): Polymorphism/Virtual function.
(CO528.5): Template, Exception.
(CO528.6): UML.

(CO532.1): To teach students how to analyze algorithms in order to determine their calculation complexity in the terms of Big Oh, Big theta and Omega. Recursions.
(CO532.2): To teach sorting algorithms (such as mergesort and quicksort) and their applications.
(CO532.3): Probabilistic Analysis and Randomized algorithms for sample problems from the following list (not limited to, and not necessarily including all): CS- Hiring, Longest Streaks, Bins and Balls problem, the Birthday paradox, and randomized quicksort.
(CO532.4): Binary search trees and optimal binary search trees, and their applications.
(CO532.5): Dynamic programming algorithms for problems such as line scheduling, matrix chain multiplication, longest common subsequence, and their practical applications.
(CO532.6): Greedy algorithms for problems such as the activity selection problem and its application to resource planning.

(CO581.2): Students will be able to describe and discuss cryptography basics.
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?

Objective#3 (PO3): Students will demonstrate the ability to recognize, design and implement efficient software solutions to problems.

Assessment will be measured through testing the following course objectives. The first percentile is Fall 2010 and the second percentile is Spring 2011.

CSCI 520 Information Structure and Algorithm Analysis.
(CO520.1): To understand the concept of sparse matrices, stack and queues.
(CO520.2): To examine the differences between linear and linked representation of stacks, queues, and ordered data.
(CO520.3): To understand and implement tree structures and to compare various sorting algorithms.

CSCI527 Advanced Databases and Data Mining
(CO527.1): Understand current status of the state-of-the-art data mining methodology in industry and academics.
(CO527.3): Learn and use effective tools for web navigation and program integration management.
(CO527.5): Construct programs for capturing association rules.
(CO527.6): Write programs for trend analysis using statistical data mining techniques.

CSCI 528 Object Oriented Methods
(CO528.2): Classes basics/advanced.
(CO528.3): Overloading.
(CO528.4): Polymorphism/Virtual function.
(CO528.5): Template, Exception.
(CO528.6): UML.

CSCI 532 Algorithm Design
(CO532.1): To teach students how to analyze algorithms in order to determine their calculation complexity in the terms of Big Oh, Big theta and Omega. Recursions.
(CO532.2): To teach sorting algorithms (such as mergesort and quicksort) and their applications.
(CO532.3): Probabilistic Analysis and Randomized algorithms for sample problems from the following list (not limited to, and not necessarily including all): CS- Hiring, Longest Streaks, Bins and Balls problem, the Birthday paradox, and randomized quicksort.
(CO532.4): Binary search trees and optimal binary search trees, and their applications.
(CO532.5): Dynamic programming algorithms for problems such as line scheduling, matrix chain multiplication, longest common subsequence, and their practical applications.
(CO532.6): Greedy algorithms for problems such as the activity selection problem and its application to resource planning.

CSCI 581 Computer and Network Security
(CO581.2): Students will be able to describe and discuss cryptography basics.
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.

**Objective #4 (PO4):** Students will demonstrate knowledge and understanding of professional ethics and responsible behavior.

Students will be able to do the following:

(CO563.1): State the basic concepts in information security, including security policies, security models, and ous security mechanisms.
(CO563.2): Understand the issues of network communications such as service, confidentiality, authentication, reliability, access control, and availability.
(CO563.3): State threats and sources of attacks in network security.

(CO581.1): Students will be able to describe and discuss information security and network security basics.

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?

**Objective #4 (PO4):** Students will demonstrate knowledge and understanding of professional ethics and responsible behavior.

Assessment will be measured through testing the following course objectives. The first percentile is Fall 2010 and the second percentile is Spring 2011.

CSCI 563 Fundamentals of Information Security & Assurance
(CO 563.1): State the basic concepts in information security, including security policies, security models, and ous security mechanisms.
(CO563.2): Understand the issues of network communications such as service, confidentiality, authentication, reliability, access control, and availability.
(CO563.3): State threats and sources of attacks in network security.

CSCI 581 Computer and Network Security
(CO581.1): Students will be able to describe and discuss information security and network security basics.
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.

Objective #5 (PO5): Students will demonstrate the ability to communicate effectively and to work as a team.

Students will be able to do the following:

CSCI 526 Database Systems  
(CO526.2): Master the technique for team play and teamwork for small scale database projects through brainstorming and joint requirement planning.  
(CO526.10): Be able to demo and present the initial, intermediate, and final delivery of the database design project.

CSCI 527 Intelligent Database Systems  
(CO527.2): Obtain the technique for team play and teamwork for large intelligent database projects through brainstorming and joint requirement planning.

CSCI 528 Object Oriented Methods  
(CO528.7): Integration Project.

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?

Objective #5 (PO5): Students will demonstrate the ability to communicate effectively and to work as a team.

Students will be able to do the following:

CSCI 526 Database Systems  
(CO526.2): Master the technique for team play and teamwork for small scale database projects through brainstorming and joint requirement planning.  
(CO526.10): Be able to demo and present the initial, intermediate, and final delivery of the database design project.

CSCI 527 Intelligent Database Systems  
(CO527.2): Obtain the technique for team play and teamwork for large intelligent database projects through brainstorming and joint requirement planning.
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.

**Objective #6 (PO6):** Students will become successful professionals able to gain employment and/or to be accepted into a Computer Science Ph.D. program.

Students will be able to do the following:

- Gain employment in the Computer Science field
- Get accepted to an academic program to continue their graduate education.
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?

**Objective #6 (PO6): Students will become successful professionals able to gain employment and/or to be accepted into a Computer Science Ph.D. program.**
Assessed by on-going follow-up surveys and letters of feedback from students.
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.

CSCI 515.002 Fundamental of Programming Fall 2011
Professor: Mutlu Mete

1. To understand the internal representation of the various data types.
2. To examine the internal representation of two and three dimension arrays in C/C++.
3. To understand dynamic memory allocation, parameter passing, the use of pointers.*

Derivation of Assessment Scores:
#1 based on Assignment 1, 2 and Test 1
#2 based on Assignment 12 and 13
#3 based on Assignment 19, Test 2, and Final

CSCI 515.001 Fundamentals of Programming Fall 2011
Associate Professor: Dan Creider

(CO515.1): To understand the internal representation of the various data types.
(CO515.2): To examine the internal representation of two and three dimension arrays in C/C++.
(CO#515.33): To understand dynamic memory allocation, parameter passing, the use of pointers.

Course: CSCI 516.001 – Fund Concepts Computing/Mach Org, Fall 2011
Professor: Nikolay Metodiev Sirakov

Objective #1 Numbering systems and conversions:
Objective #2 Intro to Computer Organization: theoretical concepts to design digital diagrams;
Objective #3 Concepts of Machine Instructions, Assembly and linking, assembly language programming, interrupts;
Objective #4 Unconditional jumps, flags, subroutines, Stacks; arithmetic, flags, registers; work with jump and loops;
Objective #5 Arrays, addressing modes and memory management, indirect addressing;
Objective #6 Advanced procedures, local variables, stack parameters, strings, link to high level language (C++);

Derivation of Assessment Scores from:
1HW; 3 In-class Problems; 4 Quizes; 2 In-class Exams, 1 Final Exam; 2 Programs, and 2 ECP

Course: 520.001 Information Structure and Algorithm Analysis Fall 2011
Associate Professor: Dan Creider

(CO520.1): To understand the concept of sparse matrices, stack and queues.
(CO520.2): To examine the differences between linear and linked representation of stacks, queues, and ordered data.
(CO520.3): To understand and implement tree structures and to compare various sorting algorithms.

Course: CSCI 520.002 Information Structures  Fall 2011
Professor: Abdullah N. Arslan

1. To understand the concept of sparse matrices, stacks, and queues
2. To examine the differences between linear and linked representation of stacks, queues and ordered data
3. To understand and implement tree structures and compare various sorting algorithms

Derivation of Assessment Scores:
#1 based on assignments 1, 5, 6, 11, quiz 2, exams 1, 2, and 3
#2 based on assignments 1, 2, 4, 5, 6, quiz 1, and 2, exams 1 and 2
#3 based on assignments 7, 8, 9, and 10, quiz3, exams 2 and 3

CSCI 525 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

Course: CSCI 526 Database Systems Fall 2011
Professor: Mutlu Mete

1. Obtain current status of the state-of-the-art database design methodology in industry and academics
2. Master the technique for team play and teamwork for small scale database projects through brain storming and joint requirement planning
3. Learn and use effective tools for logical and physical database design and development
4. Perform data normalization process for effective data management
5. Write SQL programs for effective data definition and manipulation
6. Develop ER diagrams for logical design of database systems
7. Implement a small scale database development project using commercially available DBMS tools
8. Learn to apply various data verification techniques for easy and effective data maintenance
9. Learn how to evaluate database management systems with widely-accepted industry standards
10. Be able to demo and present the initial, intermediate, and final delivery of the database design project

Derivation of Assessment Scores:
#1 based on Test 1
#2 based on Group Project
#3 based on Test 1 and Final
#4 based on Final Exam
#5 based on Assignment 2 and Final Exam
#6 based on Group Project and Test 1
#7 based on Group Project
#8 based on Group Project
#9 based on Test 1 and Final
#10 based on Group Project

Overall Assessment of Course Objectives

Course: CSCI527 (Advanced Databases and Data Mining) Fall 2011
Instructor: Sang C. Suh

[Course Objectives with assessment]
1. Understand current status of the state-of-the-art data mining methodology in industry and academics
2. Obtain the technique for team play and teamwork for large intelligent database projects through brainstorming and joint requirement planning
3. Learn and use effective tools for web navigation and program integration management
4. Identify dirty data sources and construct data cleaning programs
5. Construct programs for capturing association rules
6. Write programs for trend analysis using statistical data mining techniques
7. Implement code for generating decision rules using decision tree based classification
8. Apply divide-and-conquer approach and learn to integrate various programs of small size to form a solution to a large integrated program
9. Learn to apply various data mining techniques into various areas of different domains
10. Learn how to design a large scale software analysis and design project with a focus on business intelligence
11. Be able to demo and present the initial, intermediate, and final delivery of the system following CMM and rapid prototyping approaches

Steps being taken to better emphasize and teach objectives
1) All course objectives are successfully met.
2) Develop more supplementary course material that helps students with concept
3) Have more face-to-face interaction with each team for better mgmt of project

Derivation of Assessment Scores:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test I</td>
<td>X</td>
<td>X</td>
<td>x</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test II</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>X</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

* denotes unsuccessful objective (< 75%)

Course: CSCI 528.01W, Object Oriented Programming, Fall 2011
Instructor: Varadraj P. Gurupur, PhD

1. Software Engineering Basics
2. Classes basics/advanced
3. Overloading
4. Polymorphism/Virtual function
5. Template, Exception
6. UML
7. Integration Project

Derivation of Assessment Scores:

Objective #1 based on midterm, homework 2
Objective #2 based on homework 1
Objective #3 based on homework 2
Objective #4 based on midterm, homework 2
Objective #5 based on homework 1
Objective #6 based on homework 2 and final exam
Objective #7 based on homework 2

Course: CSCI 530.001 Operating Systems  Fall 2011
1. List and understand basic functions and parts of an OS.
2. Understand modern memory management techniques, including virtual memory.
3. Know fundamental concepts of OS such as multiprogramming and multiuser systems.
4. Understand process management algorithms, structures and threading.
5. Understand issues with concurrent and parallel programming, including deadlocks.
6. Learn specific mechanisms for modern OS such as Linux and Windows Vista.

Derivation of Assessment Scores:
#1 based on T1: 1 3 9 11 20 21; F: 1, 4, 14, 19, 27, 31
#2 based on T1: 6, 7, 13, 14, 28; T2: 4, 5, 6, 7, 9, 18, 19, 20, 21, 22, 23, 24, SP1; F: 6, 7, 11, 18, 21, 22, 24, 25, 27, 28, 34, 36, 38, 40, 41, SP1, SP2
#3 based on T1: 2, 5, 20, 23, 24, SP1; T2: 8, 25; F: 1, 3, 5, 8, 9, 10, 16, 19, 32, 33, 41
#4 based on T1: 8, 16, 18, 22, 25, 26, 27; T2: 10, 11, 12, 13, 15, 26, 27, 30, SP2; F: 2, 15, 17, 23, 26, 29, 37
#5 based on T1: 4, 10, 15, 17, 19, 29, 30; T2: 1, 2, 3, 13, 16, 17, 28; F: 12, 20, 30, 35, 39
#6 based on T1: 14, 16, 21, SP2; T2: 8, 14, 22, 25, 28, 29, SP2; F: 5, 8, 13, 16, 33

T1, T2 = first and second test questions
F = final exam test questions

Course: CSCI 532.002 Algorithm Design  Fall 2011
Professor: Abdullah N. Arslan

1. To teach students how to analyze algorithms in order to determine their computation complexity in terms of Big Oh, Big Theta and Omega. Recursions.
2. To teach sorting algorithms (such as mergesort and quicksort) and their applications.
3. Probabilistic Analysis and Randomized algorithms for problems such as randomized quicksort and Bins and Balls problem, and if time permits, CS- Hiring, Longest Streaks and the Birthday paradox.
4. Binary search trees and optimal binary search trees, and their applications.
5. Dynamic programming algorithms for problems such as line scheduling, matrix chain multiplication, longest common subsequence, and their practical applications.
6. Greedy algorithms for problems such as the activity selection problem and its application to resource planning.
7. If time permits, Graph Algorithms such as Minimum Spanning Tree algorithms and Dijkstra’s shortest path algorithm.

Derivation of Assessment Scores:
#1 based on quiz 1, exams 1 and 3
#2 based on exams 1 and 3
#3 based on exam 3
#4 based on exam 3
#5 based on exams 1 and exam 3
#6 based on quiz 2 and 3, and exams 2 and 3
#7 based on quiz 3 and exams 2 and 3
CSCI 531 Java Programming Fall 2011
Professor: Ray Maleh

(CO531.1) Code, compile and run a Java program.
(CO531.2): Master programming techniques for console input and output.
(CO531.3) Apply logical constructs for branching and loops.
(CO531.4) Define classes and methods.
(CO531.5) Create and access arrays.
(CO531.6) Develop linked data structures.
(CO531.7) Employ exception-handling programming techniques*
(CO531.8) Utilize file input and output procedures for sequential and random access.
(CO531.9) Use the Swing library to develop programs with graphical user interfaces.

Derivation of Assessment Scores
(CO531.1) based on HW 1
(CO531.2) based on Quiz 1
(CO531.3) based on Midterm
(CO531.4) based on Quiz 1, Quiz 2, and Midterm
(CO531.5) based on Quiz 1 and Midterm
(CO531.6) based on Midterm
(CO531.7) based on Midterm and Quiz 3
(CO531.8) based on Midterm
(CO531.9) based on Quiz 3 and Final Project

Course: CSCI 532.001 Algorithm Design Fall 2011
Professor: Abdullah N. Arslan

1. To teach students how to analyze algorithms in order to determine their computation complexity in terms of Big Oh, Big Theta and Omega. Recursions.
2. To teach sorting algorithms (such as mergesort and quicksort) and their applications.
3. Probabilistic Analysis and Randomized algorithms for problems such as randomized quicksort and Bins and Balls problem, and if time permits, CS- Hiring, Longest Streaks and the Birthday paradox.
4. Binary search trees and optimal binary search trees, and their applications.
5. Dynamic programming algorithms for problems such as line scheduling, matrix chain multiplication, longest common subsequence, and their practical applications.
6. Greedy algorithms for problems such as the activity selection problem and its application to resource planning.
7. If time permits, Graph Algorithms such as Minimum Spanning Tree algorithms and Dijkstra’s shortest path algorithm.

Derivation of Assessment Scores:
#1 based on quiz 1, exams 1 and 3
#2 based on exams 1 and 3
#3 based on exam 3
Objective #1: Using subnets and routing protocols, design and configure a router network.
Objective #2: Design and configure a switched network and VLANs.
Objective #3: Understand the concepts of an Access Control List and learn how to configure a router for ACLs.
Objective #4: Understand the basic concepts of a Wide Area Network and WAN components. Integrate knowledge of subnets, routers, switches, VLANs, ACLs and WANs, into an understanding of modern digital computer networks.
Objective #5: Gain practical laboratory experience working with routers and switches to implement a working network.

Derivation of percentiles:
Objective #1 is measured by semester exam #1.
Objectives #2 is measured by semester exam #2.
Objective #3 is measured by the exam #3.
Objectives #4 is measured by final exam.
Objective #5 is measured by lab grade and attendance.

Course: CSCI 538.001 Artificial Intelligence (Collective Machine Intelligence) Fall 2011
Professor: Derek Harter

1. Develop familiarity with high-level Python scripting language
2. Learn basics of fundamental machine learning techniques, such as optimization, Bayesian estimates, clustering, k-nearest neighbor, kernel methods, etc.
3. Learn basic distinction between supervised and unsupervised machine learning methods.
4. Show examples of using Web 2.0 data sources for systems development.
5. Learn basic machine learning training and testing techniques, including cross validation and data optimization.

Derivation of Assessment Scores:
#1 based on T1: 1, 2, 5, 8, 22; T2: 5, 6, 7
#2 based on T1: 3, 4, 5, 6, 15, 17; T2: 7, 8, 15, 16
#3 based on T1: 8, 9, 23, 25; T2: 1, 2, 3, 4, 10, 11
#4 based on T1: 10, 11, 12, 13, 27; T2: 20, 21, 22, 23, 24
#5 based on T1: 28, 29, 30; T2: 25, 26, 27, 28, 29, 30

T1, T2 = first and second test questions
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.

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.
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.)

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.
CSCI 270
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.