

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

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

Outcome Description

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

Assessment Method

Assessment will be measured through testing the following course objectives:

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 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

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.

CSCI 470 Database Programming

(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.

(CO470.8) Perform system and database administration to implement software to support database application development.

(CO470.9) Complete a project to implement database management software or related tools.

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

Assessment will be measured through testing the following course objectives:

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.

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

Assessment will be measured through testing the following course objectives:

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.

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

Assessment will be measured through testing the following course objectives:

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.

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.)

Program Objective #6 (PO6) : Learn common algorithms and how to analyze them for efficiency.

Assessment will be measured through testing the following course objectives:

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.

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

Assessment will be measured through testing the following course objectives:

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.

(CO428.2) Classes basics/advanced

(CO428.6) UML

CSCI 430 Operating Systems

(CO430.1) Understand the concepts, structures, and mechanisms of operating systems.

(CO430.2) Understand memory management, virtual memory, swapping, paging algorithms, segmentation, and clock paging policies.

(CO430.3) Understand multiprogramming and multiuser capabilities, and how operating systems evolved.

(CO430.4) Understand process management, process states and process and thread structures and concepts.

(CO430.5) Understand concurrent processes and associated deadlock prevention, avoidance, detection, recovery methods, and the use of semaphores.

(CO430.6) Learn specific design decisions and architectures used in modern operating systems.

CSCI 359 Systems Analysis and Design

(CO359.1) Understand concepts relating to different types of information systems.

APPENDIX A

Assessment Plan Details and Examples Graduate and Undergraduate Degree Programs Department of Computer Science & Information Systems Texas A&M University-Commerce

I. Program Embedded Assessment

The following is a description of the assessment program for the Computer Science Master's Degree Program, the Computer Science Bachelor of Science Degree, and the Computer Information Systems Bachelor of Science Degree. The assessment program contains: a) an overall comprehensive testing component; b) a course-embedded assessment component; c) a non-quantitative data gathering component.

A. Overall Comprehensive Testing Component

Currently, the Computer Science Master's program requires the successful completion of a comprehensive exam as a prerequisite for the Master's Degree. This exam is given three times a year (in the Fall, Spring, and Summer semesters). The exam is developed and administered by a committee of graduate faculty. A set of the most important course objectives are selected (see Course Embedded Assessment, Section B) for examination. Testing for these objectives are translated into a set of questions and administered. The results are translated into a percentile of those passing and failing by objective, and a list of those students passing or failing the exam. The criteria for passing is determined by the committee. Such criteria is based on the overall performance of students taking the exam. The report of the percentile of those passing or failing by objective will be used as feedback in re-evaluating techniques used in teaching these objectives.

Starting in the Spring Semester, 2002, a comprehensive exam will be also given to students graduating from the Computer Science and the Computer Information Systems Undergraduate Programs. The exam will be constructed by a committee of undergraduate faculty. It shall be a one or two hours exam and shall be given in the final days of the senior course, CSCI 440. Unlike the graduate comprehensive exam, the undergraduate comprehensive exam will not be required for graduation. It will be used only for program evaluation. The comprehensive exam will be developed and administered by a committee of undergraduate faculty. A set of the most important course objectives will be selected (see Course Embedded Assessment, Section B) for examination. Testing for these objectives will be translated into a set of questions and administered. The results will be translated into a percentile of those passing and failing by objective and will be used as feedback in re-evaluating techniques used in teaching these objectives.

Instead of using an in-house comprehensive exam, the option is reserved for a utilizing a nationally administrated exam that would nationally rank participants.

B. Course Embedded Assessment Component

The purpose of embedded assessment is to measure the degree of success in which each course objective has been met. Starting in the Spring 2003 semester, the comprehensive exam for both graduate and undergraduate students will contain embedded assessment of program objectives.

1. The most important course objectives (see Section II below) will be selected from objectives listed in core courses for both the graduate and undergraduate programs. These important objectives will comprise the overall program objectives. The selection will be made by the Graduate Curriculum Committee and Undergraduate Curriculum Committee.

2. These overall program objectives will be embedded in one of more questions given in the Graduate Comprehensive Exam and in the Undergraduate Comprehensive Exam. A passing grade on the Graduate Comprehensive Exam, as determined by the Graduate Curriculum Committee, is required for student graduation. The Undergraduate Comprehensive is not a requirement for graduation. The assessment for each objective is a percentile of the total maximum score in a similar manner that course objectives are measured. For example, if Program Objective #1 is measured by two 10 point questions and a total of 10 students take the exam, then a total maximum score for Program Objective #1 will be 200 points. If a total of 175 points is scored for Objective #1, then the overall assessment for Objective #1 is 87.5%. See Appendix A for a more detailed example.

3. Program Objectives will be divided into two categories: Category(A) 75% or greater; and Category (B) less than 75%. Objectives in Category A will be considered successful while objectives in Category B will be considered unsuccessful. The report submitted by the instructor shall include steps being taken to better emphasize and teach objectives falling into Category B.

C. Non-quantitative Data Gathering Component

A non-quantitative data gathering component shall also be part of the program assessment. This component consists of the following:

1. Exit interviews with graduating students.
2. Compilation of job positions and salaries offered to graduating students.
3. Interviews with employers of graduates.
4. Interviews with potential employers of graduates.

The purpose of these interviews are to keep informed of changes in employment patterns as we continually adjust our computer science and information systems curriculum to meet the needs of the workplace.

II. Course Embedded Assessment Plan

The purpose of embedded assessment is to measure the degree of success in which each course objective has been met. Starting in the Spring 2003 semester, selected courses taught within the Department of Computer Science & Information Systems will be designed to contain embedded assessment of course objectives on an experimental basis. Please note that while embedded course objective are similar to embedded **program objectives**, the embedded program objectives are designed to meet the requirements of accreditation. Assessment of **course objectives** are designed to spot potential problem areas within a particular course. Thus, embedded course objective assessment may be more flexible than assessment of program objectives simply because of the nature of differing course material.

Procedure for implementing this course embedded assessment plan are as follows:
(Note: this procedure plan is evaluated on a continuous basis and is subject to change.)

1. Course objectives are designated for each course taught in the department. The number of objectives should be more than 10 and probably less than 30 objectives. Multiple sections of the same course will designate identical course objectives, as agreed to by those teaching the course. On a yearly basis, all course objectives will be examined and re-evaluated by members of the Undergraduate Curriculum Committee and the Graduate Curriculum Committee. Course objectives should appear in the syllabus for that particular course. A course objective may have sub-objectives, however, only the assessment of course objectives, and not the sub-objectives, will be reported.
2. A quantitative assessment of the success of each objective shall be measured by questions given to students in regular examinations, mid-term examinations, and final examinations. This assessment measurement will exist as a percentile of the total score for each objective. (See example below).
3. Each faculty member teaching a course shall report the result of student performances of each embedded objective. The report shall be given to the department head no later than 3 weeks after the final exam. The report shall list each course objective and the percentile score of those successfully achieving this objective. Sub-objective scores need not be reported. For classes that have different instructors teaching multiple sections, the report shall be given to a designated class coordinator who will, in turn, prepare a consolidated report. The consolidated report along with each individual report shall be turned in the department head.
4. Objectives will be divided into two categories: Category(A) 75% or greater; and Category (B) less than 75%. Objectives in Category A will be considered successful while objectives in Category B will be considered unsuccessful. The report submitted by the instructor shall include steps being taken to better emphasize and teach objectives falling into Category B.

Example:

Suppose a given set of objectives for CSCI 151 are as follows:

1. Construct appropriate comments inside a C++ program.
2. Declare valid identifiers using appropriate data types in a C++ program.
3. Evaluate and construct assignment statements in a C++ program.
4. Input and output data in a C++ program.
5. Evaluate and construct selection structures using C++.
6. Evaluate and construct repetition structures using C++.
7. Construct programs consisting of multiple functions.
8. Understand the concepts of scope and lifetime.
9. Understand how and when to use value and reference parameters with functions.
10. Effectively use one dimensional arrays.
11. Create algorithmic solutions to programming problems and implement them into C++.
12. Perform the steps, without assistance, to enter, run and debug a C++ program using a C++ compiler.

Further consider that objective #4 is tested by 3 questions (10 points each) on the mid-term exam and by 2 questions (10 points each) on the final exam. Thus, a total of 50 points per student will test the achievement of objective #4. If there are 10 students in the class, then there will be a total of 500 points which measure the effectiveness of objective #4. Given the final score per student as follows:

Objective #4

Student #1 45 points
Student #2 40 points
Student #3 50 points
Student #4 10 points
Student #5 0 points
Student #6 30 points
Student #7 15 points
Student #8 25 points
Student #9 40 points
Student #10 45 points

300 points total

Percentile success rate = 300 points out of a possible 500 points = $300/500 = 60\%$

Thus we have an overall success rate of 60% with objective #4.

EXAMPLE: Given the following objectives for CSCI 151

CSCI 151 Course Objectives

1. Understand a brief history of computing, overview of programming languages and the compilation process.
2. Create algorithmic solutions to programming problems and implement them into.
3. Declare valid identifiers using appropriate data types in a program.

4. Construct and evaluate assignment statements in a program.
5. Perform data Input and output data in a program with file processing and stream I/O
6. Construct and evaluate selection structures.
7. Construct and evaluate repetition structures.
8. Construct programs consisting of multiple functions.
9. Understand the concepts of scope and lifetime.
10. Understand how and when to use value and reference parameters with functions.
11. Effectively use one dimensional arrays.
12. Construct appropriate comments/documentation to a program.
13. Perform the steps, without assistance, to enter, run and debug a program using an IDE.

Suppose the evaluation for Objective #1 was embedded in two questions in exam #1 (20 points) and in one question in the final exam (10 points). Thus the total number of points used for the assessment of Objective #1 is 30 points. There are 21 students in the class or a total of 630 points for the overall assessment of Objective #1. Further suppose that students achieved a grand total of only 300 points on the two questions in exam #1 and on the one question in the final exam. Thus the overall assessment of Objective #1 is 47.6%. This overall assessment will be noted in a report to the Department Head.

Overall Assessment of Objectives

Course: CSCI 151.001 Spring 2002

Instructor: Will McWhorter

- **1. Understand a brief history of computing, overview of programming languages and the compilation process.
- 2. Create algorithmic solutions to programming problems and implement them into.
- **3. Declare valid identifiers using appropriate data types in a program.
- **4. Construct and evaluate assignment statements in a program.
- **5. Perform data Input and output data in a program with file processing and stream I/O
- 6. Construct and evaluate selection structures.
- 7. Construct and evaluate repetition structures.
- 8. Construct programs consisting of multiple functions.
- 9. Understand the concepts of scope and lifetime.
- 10. Understand how and when to use value and reference parameters with functions.
- **11. Effectively use one dimensional arrays.
- 12. Construct appropriate comments/documentation to a program.
- 13. Perform the steps, without assistance, to enter, run and debug a program using an IDE.

** denotes unsuccessful objectives (< 75%)

Objectives	Percentile Achievement
1)	48%
2)	88%
3)	20%
4)	58%
5)	67%
6)	98%
7)	78%
8)	88%
9)	90%
10)	88%
11)	69%
12)	99%
13)	98%

Category A (Successful \geq 75%)

2 6 7 8 9 10 12 13

Category B (Unsuccessful < 75%)

1 3 4 5 11